

Document de formation
pour une solution complète d'automatisation
Totally Integrated Automation (T I A)

MODULE A3




Initiation à la programmation d'API avec STEP 7

Ce document a été édité par Siemens A&D SCE (Automatisierungs– und Antriebstechnik, Siemens A&D Cooperates with Education) à des fins de formation.
Siemens ne se porte pas garant de son contenu.

La communication, la distribution et l'utilisation de ce document sont autorisées dans le cadre de formation publique. En dehors de ces conditions, une autorisation écrite par Siemens A&D SCE est exigée (M. Knust: E-Mail: michael.knust@hvr.siemens.de).

Tout non-respect de cette règle entraînera des dommages et intérêts. Tous les droits, ceux de la traduction y compris, sont réservés, en particulier dans le cas de brevets ou de modèles déposés.

Nous remercions l'entreprise Michael Dziallas Engineering et les enseignants d'écoles professionnelles ainsi que tous ceux qui ont participé à l'élaboration de ce document.

1.		Avant-propos.....	5
2.		Conseils pour la programmation du SIMATIC S7-300 avec STEP 7.....	7
2.1		Système d'automatisation SIMATIC S7-300.....	7
2.2		Logiciel de programmation STEP 7.....	7
3.		Installation du logiciel STEP 7.....	8
4.		Paramétrage de l'interface de programmation (PC Adapter).....	9
5.		Qu'est ce qu'un automate et à quoi sert-il ?	14
5.1		Que signifie le concept automate programmable ?.....	14
5.2		Comment l'automate programmable commande-t-il le processus ?	14
5.3		Comment l'automate reçoit-il les informations sur les états du processus ?.....	15
5.4		Quelle est la différence entre les contacts à ouverture et à fermeture ?.....	15
5.5		Comment l'automate adresse-t-il des signaux d'entrée/sortie ?	16
5.6		Comment le programme est-il traité dans l'automate ?.....	17
5.7		A quoi ressemblent les opérations logiques dans le programme de l'automate ? ...	18
5.7.1		ET logique.....	18
5.7.2		OU logique	20
5.7.3		Négation.....	21
5.8		Comment le programme d'automatisation est-il créé ? Comment arrive-t-il dans la mémoire de l'automate programmable ?.....	22
6.		Installation et mise en service du SIMATIC S7-300	23
7.		Exemple d'application.....	26
8.		Création du projet STEP 7	27
9.		Ecriture du programme STEP 7 dans le logigramme LOG.....	35
10.		Test du programme STEP 7 dans la CPU.....	47

Les symboles suivants seront utilisés dans ce module :



Information



Installation



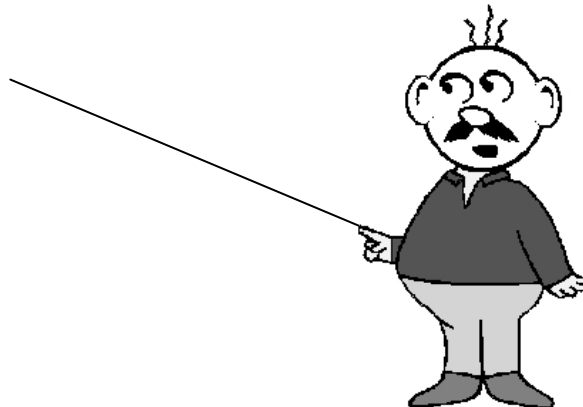
Programmation



Exemple d'application

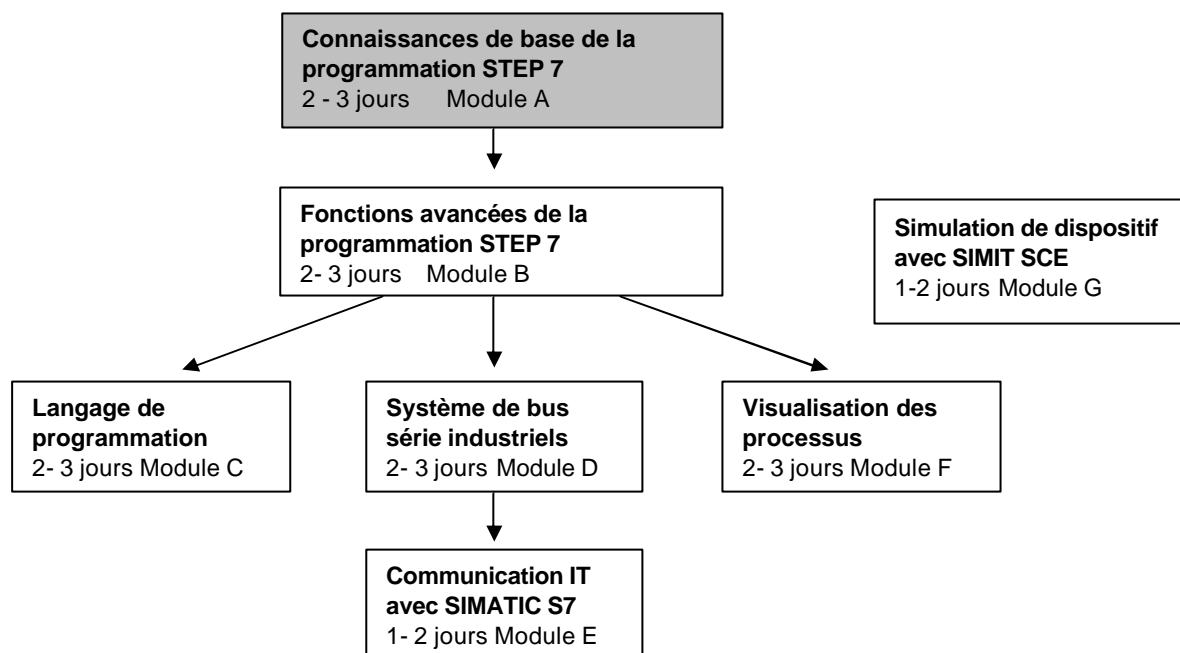


Indications



1. AVANT-PROPOS

Le contenu du module A3 est assigné à l'unité **‚Connaissances de base de la programmation STEP 7’** et présente une **rapide entrée en matière** dans la programmation STEP 7.



Objectif :

Dans ce module, le lecteur va apprendre comment programmer un automate programmable à l'aide de l'outil STEP 7. Ce cours a pour but de vous donner les notions de base et présente la manière de procéder à travers un exemple détaillé.

- Installation du logiciel et paramétrage de l'interface de programmation
- Explication de ce qu'est un automate programmable et de son fonctionnement
- Installation et mise en service d'un automate SIMATIC S7-300
- Création d'un programme-exemple
- Chargement et test du programme-exemple

Pré-requis :

Les connaissances suivantes sont requises pour l'étude de ce module :

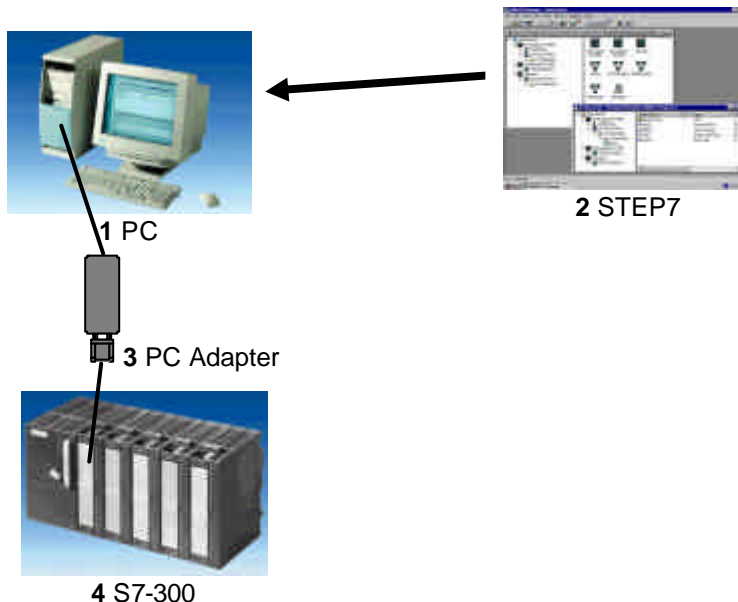
- Systèmes d'exploitation : Windows 95/98/2000/ME/NT4.0/XP

Configurations matérielles et logicielles requises

- 1 PC, système d'exploitation : Windows 95/98/2000/ME/NT4.0/XP avec
 - Minimum : 133MHz et 64Mo RAM, 65 Mo d'espace disponible
 - Optimal : 500MHz et 128Mo RAM, 65 Mo d'espace disponible
- 2 Logiciel STEP7 V 5.x
- 3 Interface ordinateur MPI (Ex : PC Adapter)
- 4 Automate SIMATIC S7-300 avec au moins un module d'entrées/sorties numérique. Les entrées doivent sortir sur le tableau de commande.

Exemple de configuration :

- Bloc d'alimentation : PS 307 2A
- CPU : CPU 314
- Entrées numériques : DI 16x DC24V
- Sorties numériques : DO 16x DC24V / 0,5 A



2. CONSEILS POUR LA PROGRAMMATION DU SIMATIC S7-300 AVEC STEP 7

2.1 SYSTEME D'AUTOMATISATION SIMATIC S7-300



Le système d'automatisation SIMATIC S7-300 est un automate modulaire compact pour une gamme de compétence inférieure et moyenne.

Vous trouvez une gamme étendue de modules S7-300 pour répondre de manière optimale à votre tâche d'automatisation.

L'automate S7 est constitué d'une alimentation, d'une CPU et d'un module d'entrées ou de sorties (Modules E/S). A ceux-ci peuvent s'ajouter des processeurs de communication et des modules de fonction qui se chargeront de fonctions spéciales, telles que la commande d'un moteur pas à pas par exemple.

L'automate programmable contrôle et commande une machine ou un processus à l'aide du programme S7. Les modules d'entrées/sorties sont adressés dans le programme S7 via les adresses d'entrée (E) et adresses de sortie (S).

L'automate est programmé à l'aide du logiciel STEP 7.

2.2 LOGICIEL DE PROGRAMMATION STEP 7



Le logiciel STEP 7 est l'outil de programmation des systèmes d'automatisation :

- SIMATIC S7-300
- SIMATIC S7-400
- SIMATIC WinAC

STEP 7 offre les fonctions suivantes pour l'automatisation d'une installation :

- Configuration et paramétrage du matériel
- Paramétrage de la communication
- Programmation
- Test, mise en service et maintenance
- Documentation, archivage
- Fonctions de diagnostic et d'exploitation

Vous trouvez pour toutes ces fonctions une aide en ligne détaillée.

3. INSTALLATION DU LOGICIEL STEP 7



STEP 7 existe en trois variantes :

- **La version de base STEP 7** permet l'utilisation d'autres logiciels optionnels tels que S7-GRAPH ou S7-PLCSIM. Cet ensemble de logiciel requiert une autorisation.
- **Le logiciel STEP 7 de base pour les étudiants** comporte un logiciel optionnel S7-PLCSIM. Ce logiciel nécessite une autorisation et elle est valable pour 120 jours.
- **La version STEP 7 Mini** (n'est plus disponible !) en tant que version pour débutant. Aucune autorisation n'est requise. Cette version ne permet pas l'utilisation d'autres logiciels tels que S7-PLCSIM ou S7-GRAPH.

STEP 7 est livré sur Cd-rom, avec une disquette pour la version de base de STEP 7 contenant l'autorisation qui devra être transférée au PC et permet l'utilisation de la version de base de STEP 7. Cette autorisation peut être retransférée sur la disquette pour être utilisée sur un autre PC. Pour en savoir plus sur l'installation et le transfert des autorisations, veuillez vous référer au module 2 – Installation de STEP 7 V5.x / Manipulation de l'autorisation.

Pour installer STEP 7, procédez comme suit :

1. Insérez le CD de STEP 7 dans le lecteur de CD-ROM.
2. Le programme est automatiquement lancé. Si ce n'est pas le cas, effectuez un double clic sur le fichier → '**setup.exe**'.
Le programme d'installation vous guide tout au long de l'installation de STEP 7.
3. Une autorisation, c'est-à-dire une licence, est requise pour l'utilisation de la version de base de STEP 7 sur votre ordinateur. Vous devez transférer celle-ci de la disquette d'autorisation sur l'ordinateur.
Ceci a lieu à la fin de l'installation. Le programme d'installation vous demande dans une boîte de dialogue si vous souhaitez procéder à l'installation de l'autorisation. Si vous cliquez sur '**Oui**', vous êtes ensuite invité à insérer la disquette d'autorisation dans le lecteur pour transférer l'autorisation sur le lecteur.

4. PARAMETRAGE DE L'INTERFACE DE PROGRAMMATION (PC ADAPTER)



Une liaison MPI est nécessaire pour programmer un SIMATIC S7-300 depuis le PC ou la PG. MPI signifie **M**ulti **P**oint **I**nterface (interface multipoint) et est une interface de communication utilisée pour la programmation, le contrôle-commande avec HMI et l'échange de données entre des CPU SIMATIC S7 jusqu'à 32 nœuds maximum.

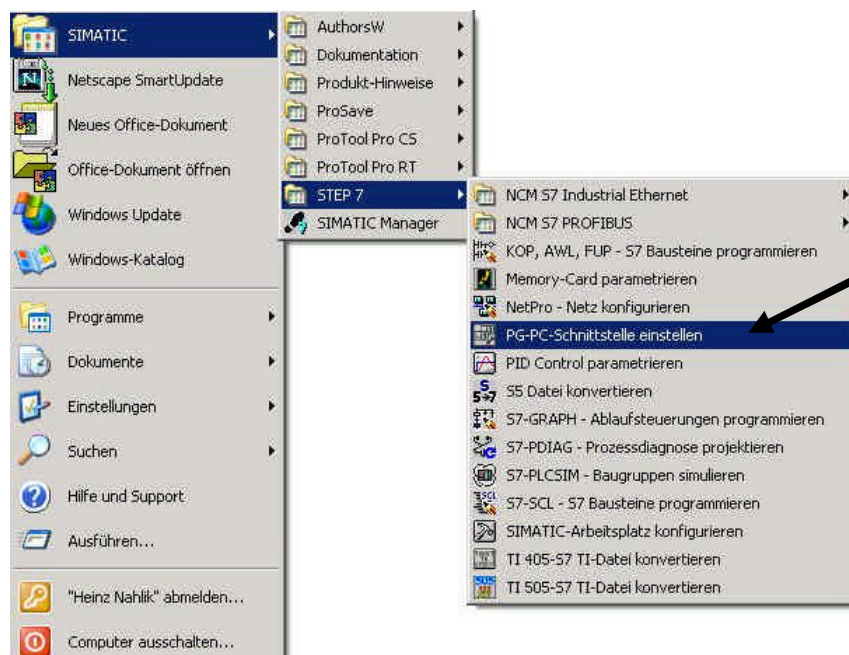
Chaque CPU du SIMATIC S7-300 est équipée d'une telle interface intégrée.

Il existe plusieurs possibilités pour raccorder le PC, la PG ou un portable à l'interface MPI :

- Processeurs de communication ISA intégrés pour la PG
- Processeurs de communication ISA pour le PC (Ex : carte MPI-ISA)
- Processeurs de communication PCI pour le PC (Ex : CP5611)
- Processeurs de communication PCMCIA pour le portable (Ex : CP5511)
- Adaptateur pour la communication via l'interface série du PC ou du portable (Ex : PC Adapter)

Nous décrivons par la suite étape par étape comment installer et paramétrer le PC Adapter pour le PC.

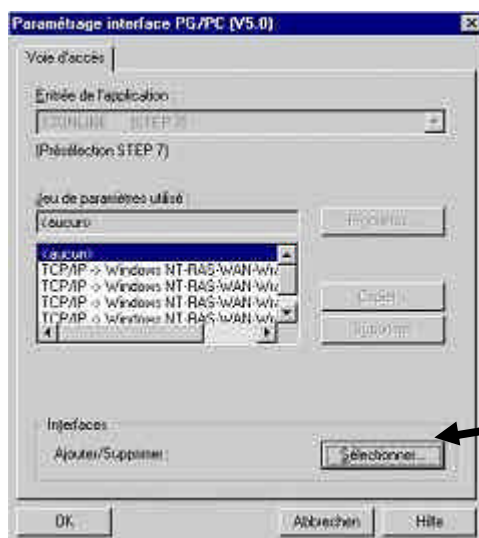
1. Appelez le '**Paramétrage de l'interface PG/PC**'. (→ Démarrer → SIMATIC → STEP7 → Paramétrage de l'interface PG/PC)



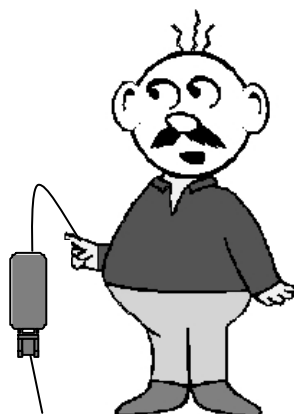
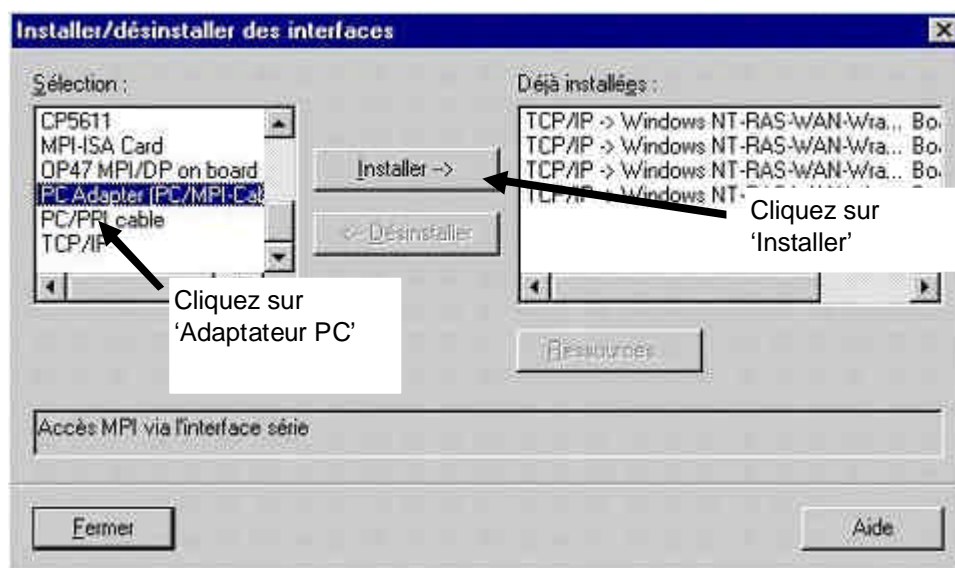
Cliquez sur
'Paramétrage
de l'interface
PG/PC'



2. Cliquez sur le bouton **,'Sélectionner'** pour installer l'interface MPI proposée. (→ Sélectionner)

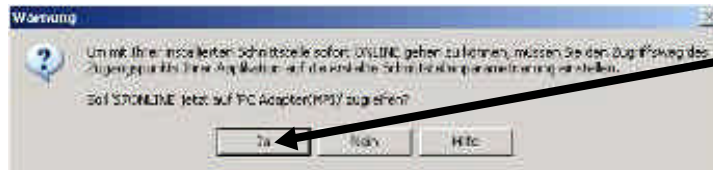


3. Sélectionnez la carte voulue, par exemple **'PC Adapter'**, et choisissez **'Installer'** (PC Adapter → Installer).

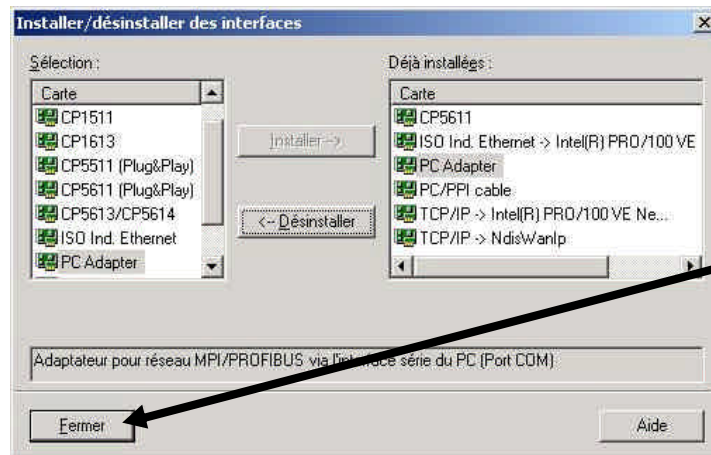




- Validez avec **,Oui'**. Dans le cas d'une nouvel enregistrement d'interface de STEP 7 en tant qu'interface standard, veuillez utiliser l'accès **,en ligne'**. Ensuite **,fermez'** la boîte de dialogue pour le choix de l'interface. (→ Oui → Fermer)

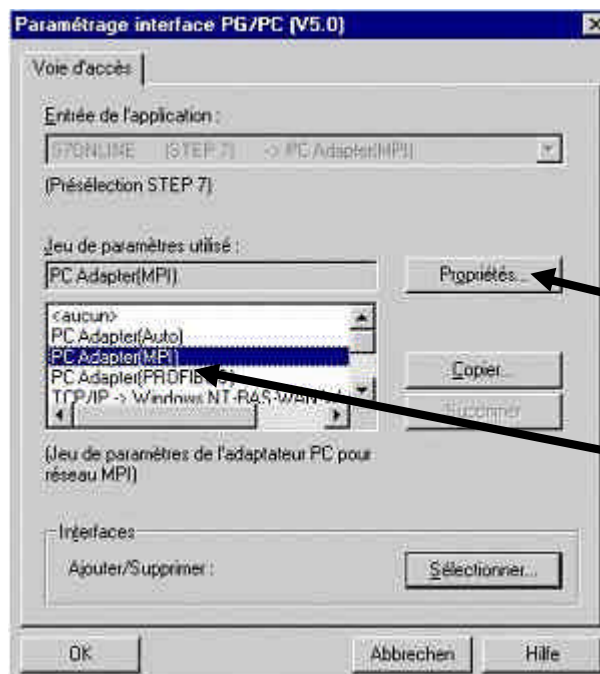


Cliquez sur 'Oui'



Cliquez sur 'Fermer'

- Choisissez les **'Propriétés'** du **'PC Adapter (MPI)'** (→ PC Adapter (MPI) → Propriétés)

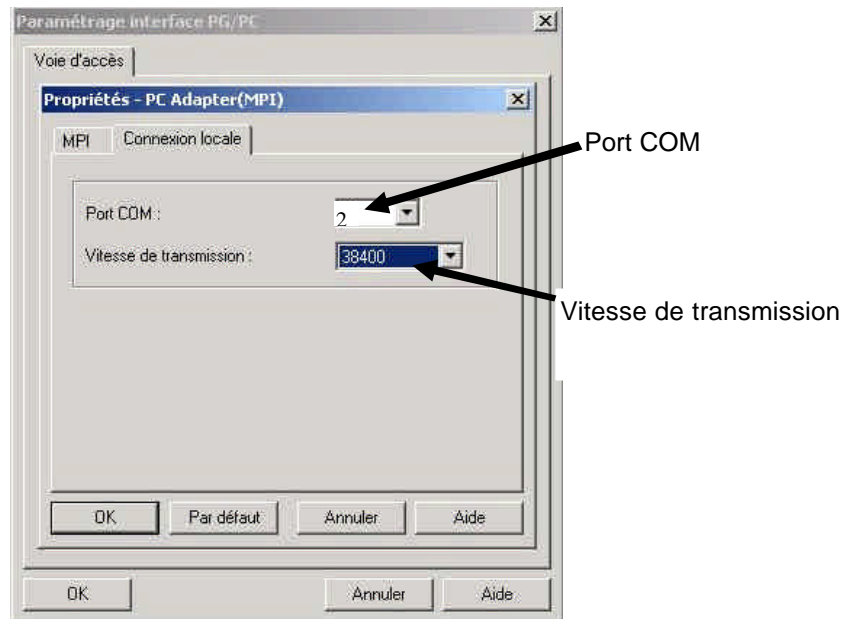


Cliquez sur 'Propriétés'

Cliquez sur 'Adapter PC (MPI)'

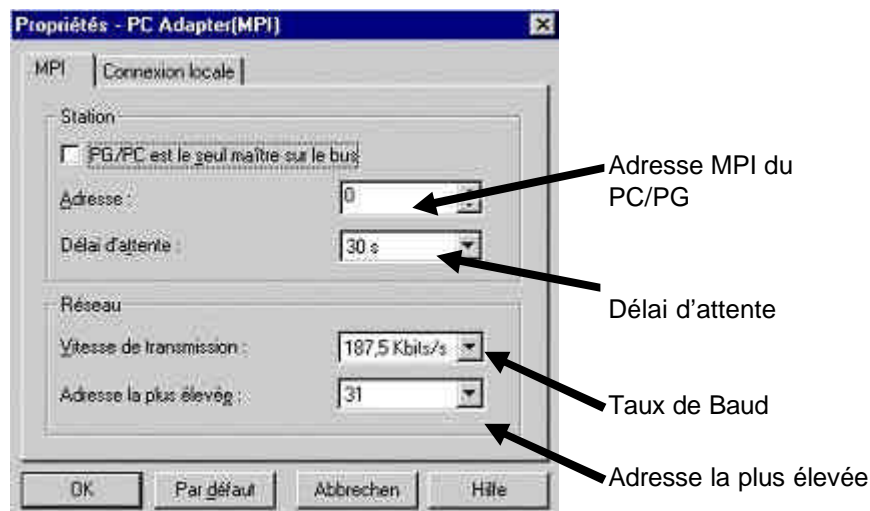


6. Réglez le '**Port COM**' et la '**Vitesse de transmission**' de l'interface série.



Indications : La vitesse de transmission doit être également réglée sur le PC Adapter ! Les adaptateurs PC de l'ancienne génération (aussi appelés câbles PC/MPI) ne peuvent traiter qu'une vitesse de transmission plus lente de 19200 bit/s.

7. Sélectionnez des valeurs pour '**Adresse MPI**', '**Délai d'attente**', '**Vitesse de transmission**' et '**Adresse la plus élevée**'.



Indications : Il est recommandé de laisser les valeurs par défaut !

8. Validez les paramètres (→ OK → OK).



9. Pour voir si le paramétrage est correct, appelez maintenant '**SIMATIC Manager**' en double-cliquant sur son icône. (→ SIMATIC Manager).



SIMATIC Manager

10. Enfichez ensuite le connecteur provenant de l'interface MPI du PC dans l'interface MPI de la CPU et activez l'alimentation en courant de l'automate. L'interface MPI est située derrière le couvercle de la CPU. Il s'agit d'un connecteur sub D à 9 points.
11. Cliquez ensuite sur le bouton  – **Partenaires accessibles**'. Si tous les paramètres sont corrects, l'image suivante s'affiche avec le répertoire des partenaires MPI accessibles. Vous pouvez également lire ici l'adresse MPI de la CPU connectée, qui est à l'adresse par défaut 2. Si plusieurs automates sont connectés ou si des CP/FM sont branchés sur l'automate, alors il apparaîtra ici plusieurs adresse MPI. (→ )

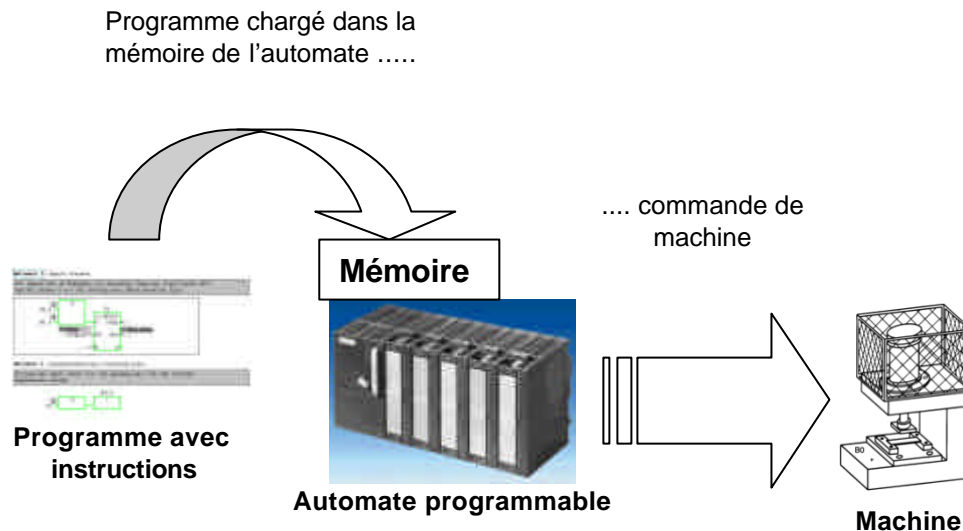


5. QU'EST-CE QU'UN AUTOMATE ET A QUOI SERT-IL ?

5.1 QUE SIGNIFIE LE CONCEPT AUTOMATE PROGRAMMABLE ?



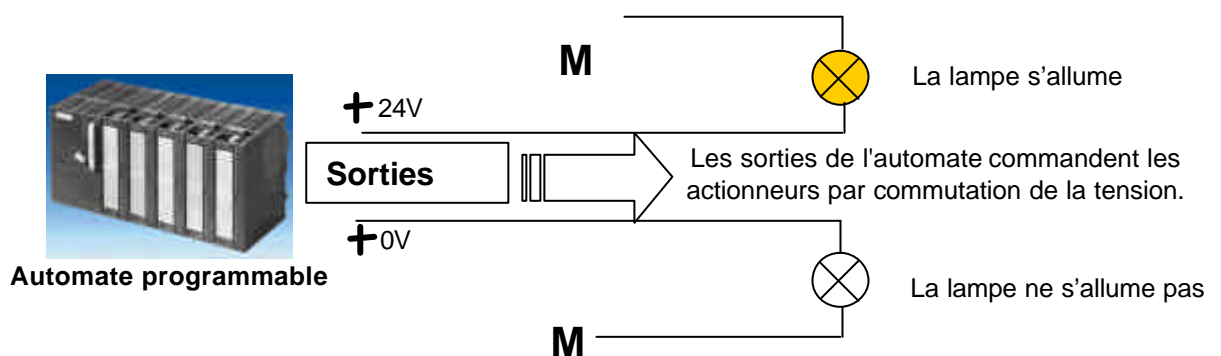
L'**automate programmable** ou **système d'automatisation** est un appareil qui commande un processus (par exemple une machine à imprimer pour l'impression de journaux, une installation de remplissage de ciment, une presse pour le moulage de formes plastiques sous pression, etc.). Ceci est possible grâce aux instructions d'un programme stocké dans la mémoire de l'appareil.



5.2 COMMENT L'AUTOMATE COMMANDE-T-IL LE PROCESSUS ?



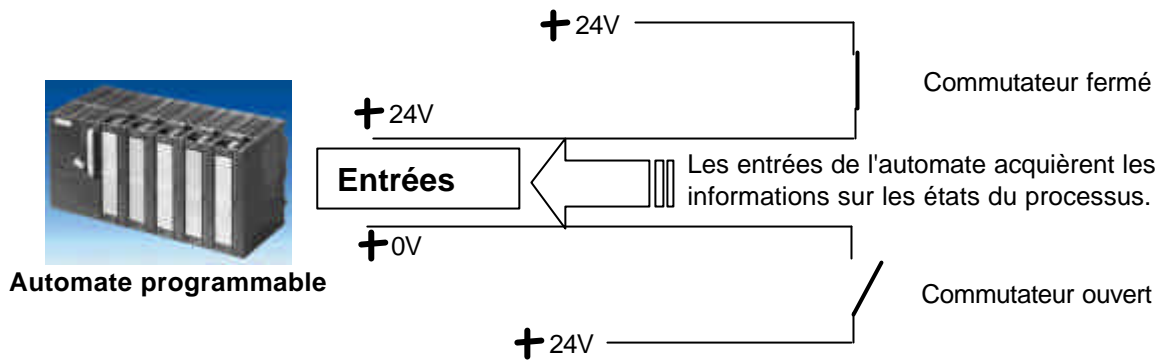
L'automate commande le processus en appliquant une tension de 24V, par exemple, aux **actionneurs** via les points de connexion de l'automate appelés **sorties**. Ceci permet d'activer ou de désactiver des moteurs, de faire monter ou descendre des électrovannes ou d'allumer ou éteindre des lampes.



5.3 COMMENT L'AUTOMATE REÇOIT-IL LES INFORMATIONS SUR LES ETATS DU PROCESSUS?



L'automate programmable reçoit les informations sur le processus via les **capteurs de signaux** reliés aux **entrées** de l'automate. Ces capteurs de signaux peuvent, par exemple, être des capteurs qui reconnaîtront si la pièce d'usinage se trouve à une position donnée ou de simples commutateurs ou interrupteurs qui peuvent être fermés ou ouverts. Il est également fait la distinction entre les **contacts à ouverture** qui sont fermés au repos et les **contacts à fermeture** qui sont ouverts au repos.

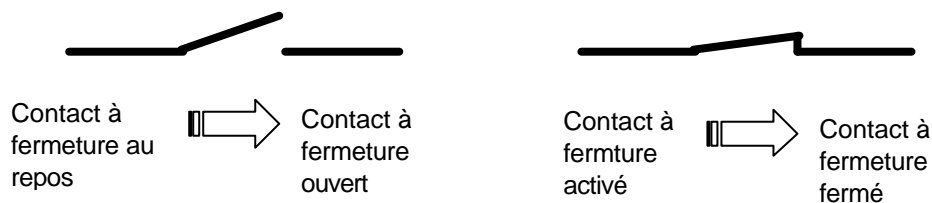


5.4 QUELLE EST LA DIFFERENCE ENTRE LES CONTACTS A OUVERTURE ET A FERMETURE ?

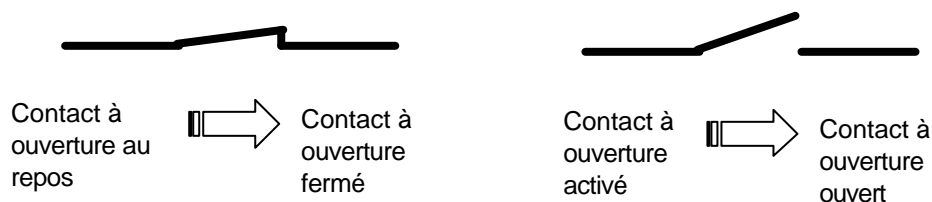


On distingue parmi les capteurs de signaux les **contacts à ouverture** et les **contacts à fermeture**.

Le commutateur ci-dessous est un contact à fermeture qui se ferme lorsqu'il est activé.



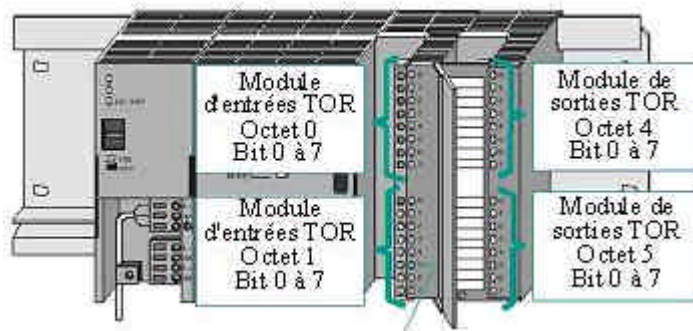
Le commutateur ci-dessous est un contact à ouverture qui est fermé quand il n'est pas activé.



5.5 COMMENT L'AUTOMATE ADRESSE-T-IL DES SIGNAUX D'ENTREE/SORTIE ?



La déclaration d'une entrée ou sortie donnée à l'intérieur d'un programme s'appelle l'adressage. Les entrées et sorties des automates sont la plupart du temps regroupées en groupes de huit entrées ou sorties numériques. Cette unité de huit entrées ou sorties est appelée un **octet**. Chaque groupe reçoit un numéro que l'on appelle l'**adresse d'octet**. Afin de permettre l'adressage d'une entrée ou sortie à l'intérieur d'un octet, chaque octet est divisé en huit **bits**. Ces derniers sont numérotés de 0 à 7. On obtient ainsi l'**adresse du bit**. L'automate programmable représenté ici a les octets d'entrée 0 et 1 ainsi que les octets de sortie 4 et 5.



Pour adresser par exemple la cinquième entrée à partir du haut, il faut entrer l'adresse suivante :

E 0 . 4

E désigne le type de l'adresse : Entrée, **0** l'adresse de l'octet et **4** l'adresse du bit.
L'adresse de l'octet et l'adresse du bit sont toujours séparées par un point.



Indications : L'adresse du bit de la cinquième entrée est un **4** car la numérotation commence à zéro.

Pour adresser par exemple la dernière sortie, il faut entrer l'adresse suivante :

A 5 . 7

A désigne le type de l'adresse : Sortie, **5** l'adresse de l'octet et **7** l'adresse de bit.
L'adresse de l'octet et l'adresse du bit sont toujours séparées par un point.



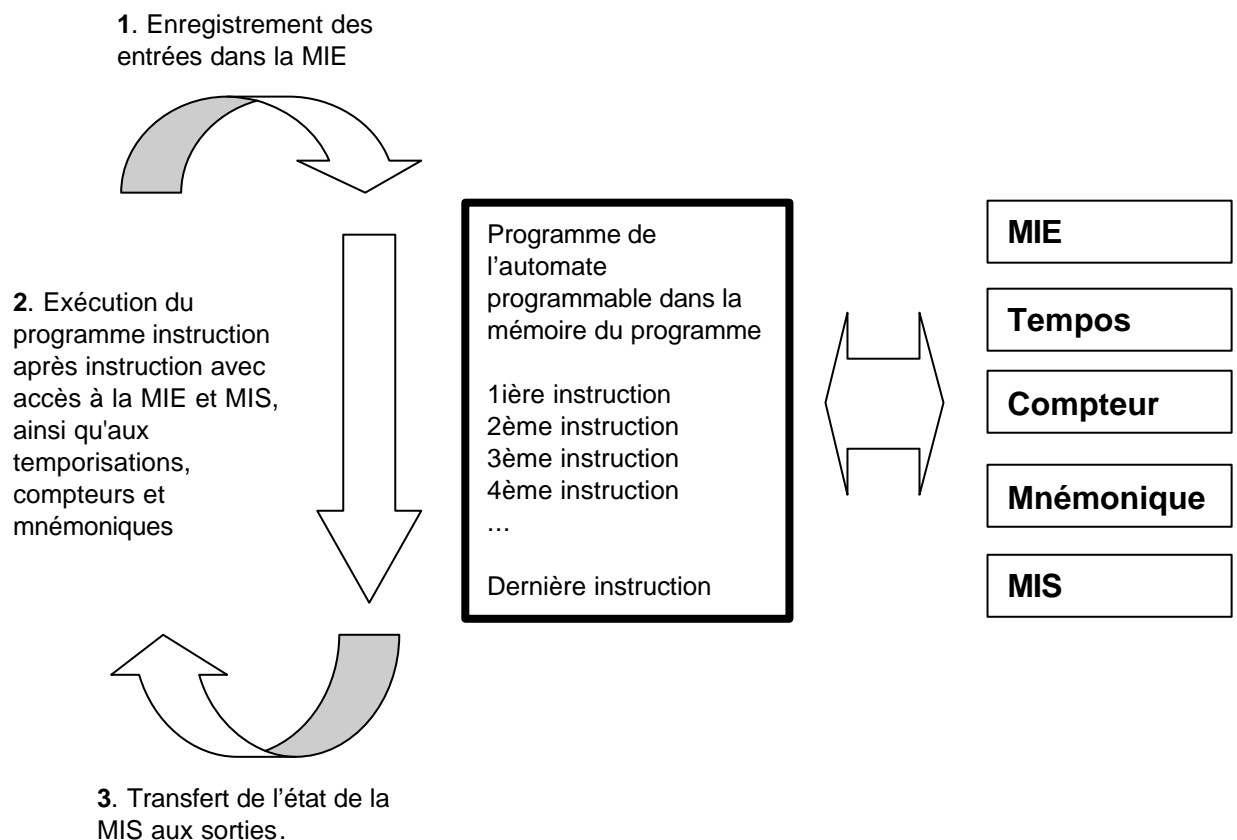
Indications: L'adresse du bit de la dernière sortie est un **7** car la numérotation commence à zéro.

5.6 COMMENT LE PROGRAMME EST-IL TRAITE DANS L'AUTOMATE ?



Le traitement du programme dans l'automate est cyclique et se déroule comme suit :

1. Après la mise sous tension de l'automate programmable, le **processeur** qui constitue pour ainsi dire le cerveau de l'automate vérifie si chaque entrée est sous tension ou non. L'état de ces entrées est enregistré dans la mémoire image des entrées (**MIE**). Si l'entrée est sous tension, l'information 1 ou "High" sera enregistrée. Si l'entrée n'est pas sous tension, l'information 0 ou "Low" sera enregistrée.
2. Ce processeur exécute le programme stocké en mémoire de programme. Celui-ci est constitué d'une liste d'instructions et d'opérations logiques exécutées de manière séquentielle. L'information d'entrée requise à cet effet est prélevée dans la mémoire image des entrées lue auparavant et les résultats logiques sont écrits dans une mémoire image des sorties (**MIS**). Durant l'exécution du programme le processeur accède également aux zones de mémoire des compteurs, temporisations et mnémoniques.
3. Dans la troisième étape, l'état est transmis après l'exécution du programme utilisateur de la MIS aux sorties, activant ou désactivant celles-ci. L'exécution du programme revient ensuite au point 1.



Indications : Le temps requis par le processeur pour l'exécution du programme s'appelle le temps de cycle. Ce dernier dépend entre autres du nombre et du type d'instructions.

5.7 A QUOI RESSEMBLENT LES OPERATIONS LOGIQUES DANS LE PROGRAMME DE L'AUTOMATE ?



Les opérations logiques servent à définir des conditions pour l'activation d'une sortie. Elles peuvent être créées dans le programme de l'automate programmable dans les langages de programmation Schéma des circuits (**CONT**), Logigramme (**LOG**) ou Liste d'instructions (**LIST**). Nous nous limiterons en vue de simplification ici au langage **LOG**. Il existe de nombreuses opérations logiques pouvant être mises en œuvre dans des programmes d'automatisation. L'opération **ET** et l'opération **OU** ou bien la **NEGATION** d'une entrée sont les opérations les plus fréquemment utilisées et seront expliquées ici à l'appui d'un exemple.

Indications : Pour obtenir rapidement et efficacement des informations sur les autres opérations logiques, consultez l'aide en ligne.

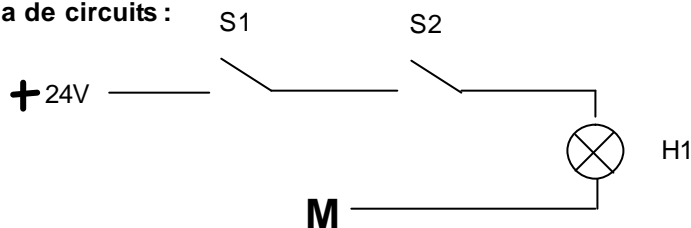
5.7.1 ET LOGIQUE



Exemple d'une opération ET :

La lampe doit s'allumer après l'activation simultanée de deux contacts à fermeture.

Schéma de circuits :



Explications :

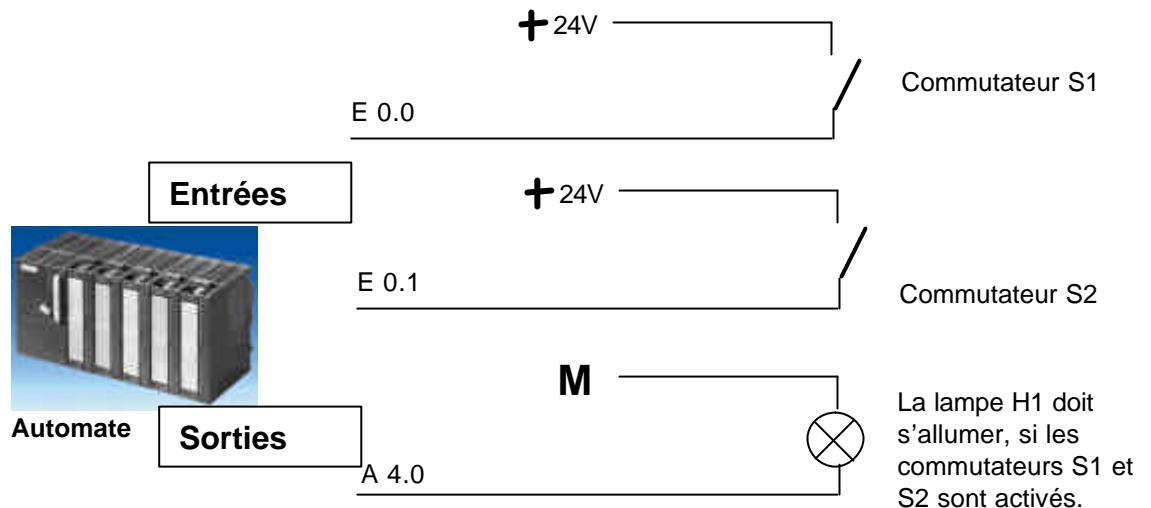
La lampe H1 s'allume au moment précis où les deux commutateurs sont activés. Si les commutateurs S1 **et** S2 sont activés, alors la lampe H1 s'allume.



Circuit de protection de l'automate programmable :

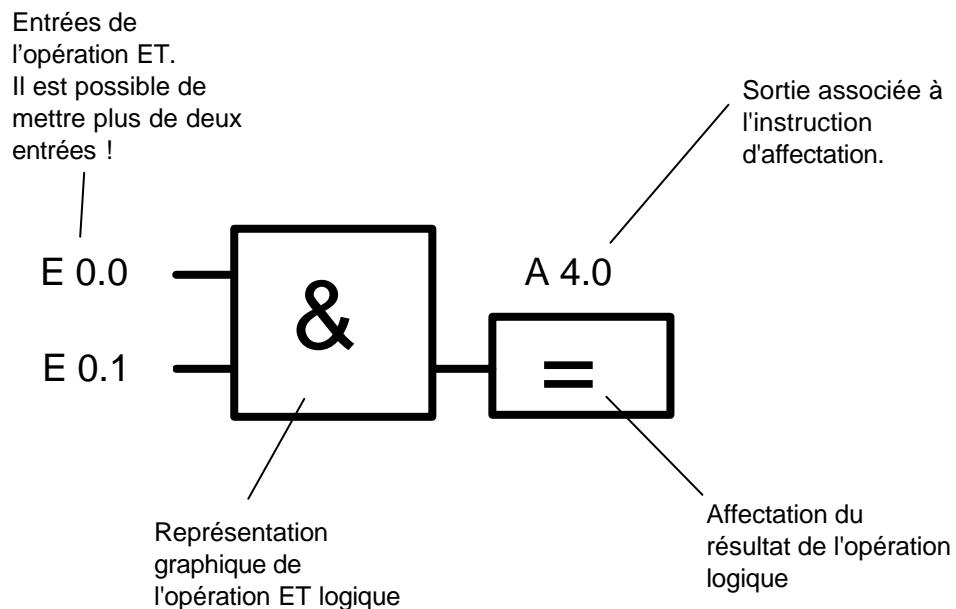
Pour que cette logique puisse être réalisée dans le programme d'automatisation, les deux commutateurs doivent naturellement être reliés aux entrées de l'automate programmable. S1 est relié ici à l'entrée E 0.0 et S2 à l'entrée E 0.1.

De plus, la lampe H1 doit être reliée à une sortie par exemple A 4.0



Opération ET dans LOG :

L'opération ET est programmée dans le Logigramme (LOG) par le symbole ci-dessous et est représentée de la manière suivante :



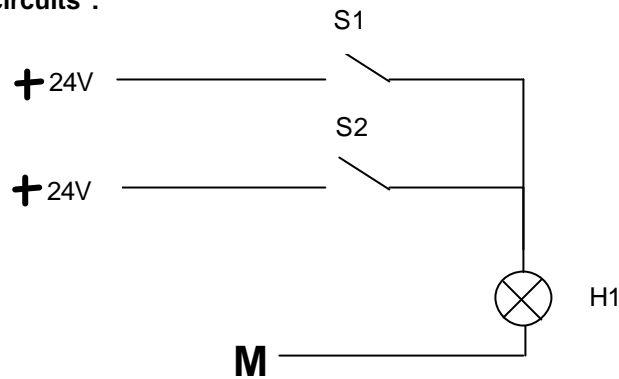
5.7.2 OU LOGIQUE



Exemple d'une opération OU :

Une lampe doit s'allumer, si l'un ou les deux contacts à fermeture sont activés.

Schéma de circuits :



Explication :

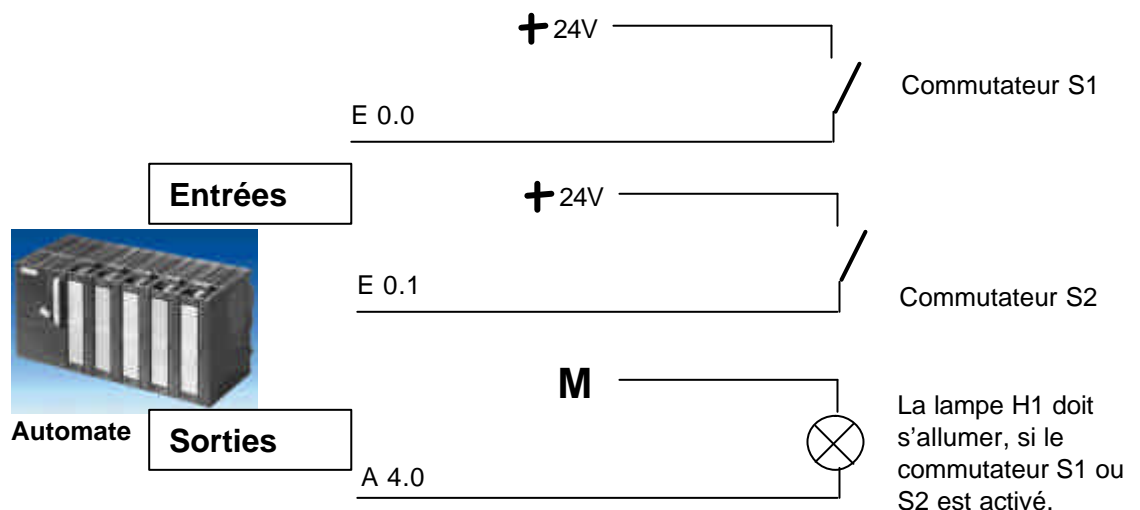
La lampe s'allume au moment précis où l'un ou les deux commutateurs sont activés.

Si le commutateur S1 **ou** S2 est activé, alors la lampe H1 s'allume.

Circuit de protection de l'automate programmable :

Pour que cette logique puisse être réalisée dans le programme d'automatisation, les deux commutateurs doivent naturellement être reliés aux entrées de l'automate programmable. Ici S1 est relié à l'entrée E 0.0 et S2 à l'entrée E 0.1.

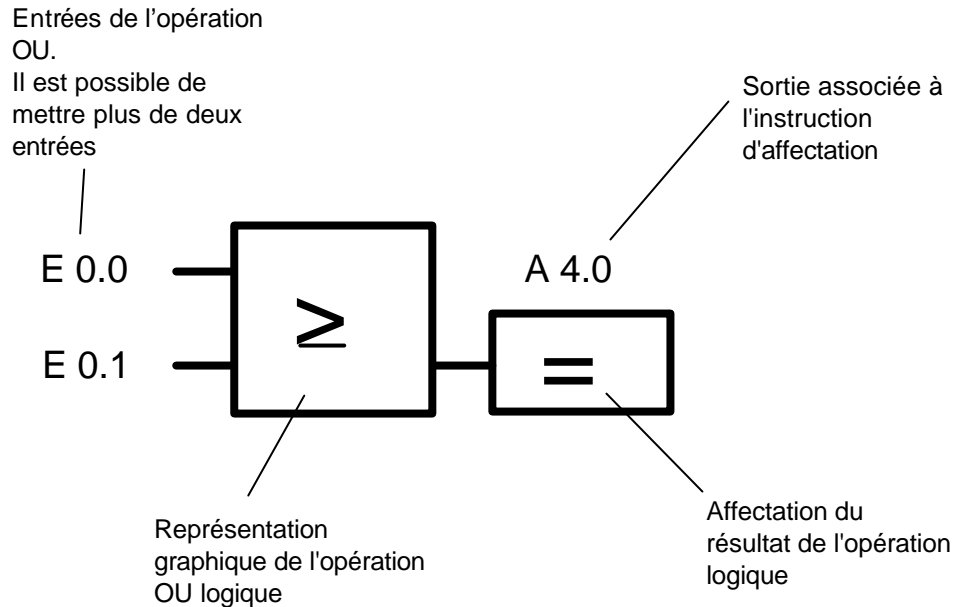
De plus, la lampe H1 doit être reliée à une sortie par exemple A 4.0.





Opération OU dans LOG :

L'opération OU est programmée dans le Logigramme (LOG) par le symbole ci-dessous et est représentée de la manière suivante :



5.7.3 NEGATION

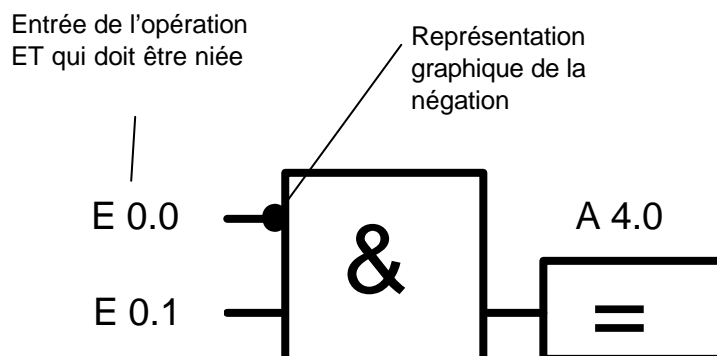


Il est souvent nécessaire dans les opérations logiques d'interroger l'état d'un contact pour savoir :

- dans le cas d'un **contact à fermeture** si celui-ci **n'a pas été activé** ou
- dans le cas d'un **contact à ouverture** s'il **a été activé**

et donc pour savoir si la tension est appliquée à la sortie ou non.
Ceci peut être réalisé par la programmation d'une **négation** à l'entrée de l'opération ET ou OU.

La négation d'une entrée de l'opération ET est programmée dans le Logigramme (LOG) par le symbole ci-dessous et est représentée de la manière suivante :



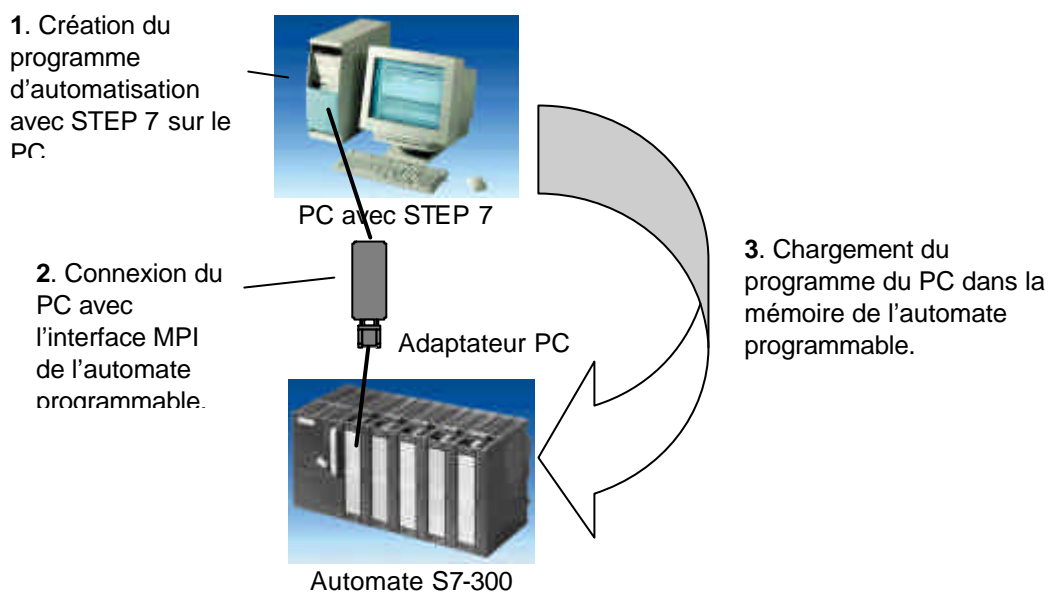
La tension est appliquée à la sortie A 4.0, si E 0.0 est désactivée et E 0.1 activée.

5.8 COMMENT LE PROGRAMME D'AUTOMATISATION EST-IL CREE ? COMMENT ARRIVE-T-IL DANS LA MEMOIRE DE L'AUTOMATE PROGRAMMABLE ?



Le programme de l'automate programmable est créé sur le PC avec le logiciel STEP 7 et y est temporairement enregistré.

Une fois que vous avez relié le PC avec l'interface MPI de l'automate, le programme peut alors être chargé à l'aide de la fonction de chargement dans la mémoire de l'automate programmable.



Indications : Cette procédure est décrite étape par étape dans les chapitres 8 à 10.

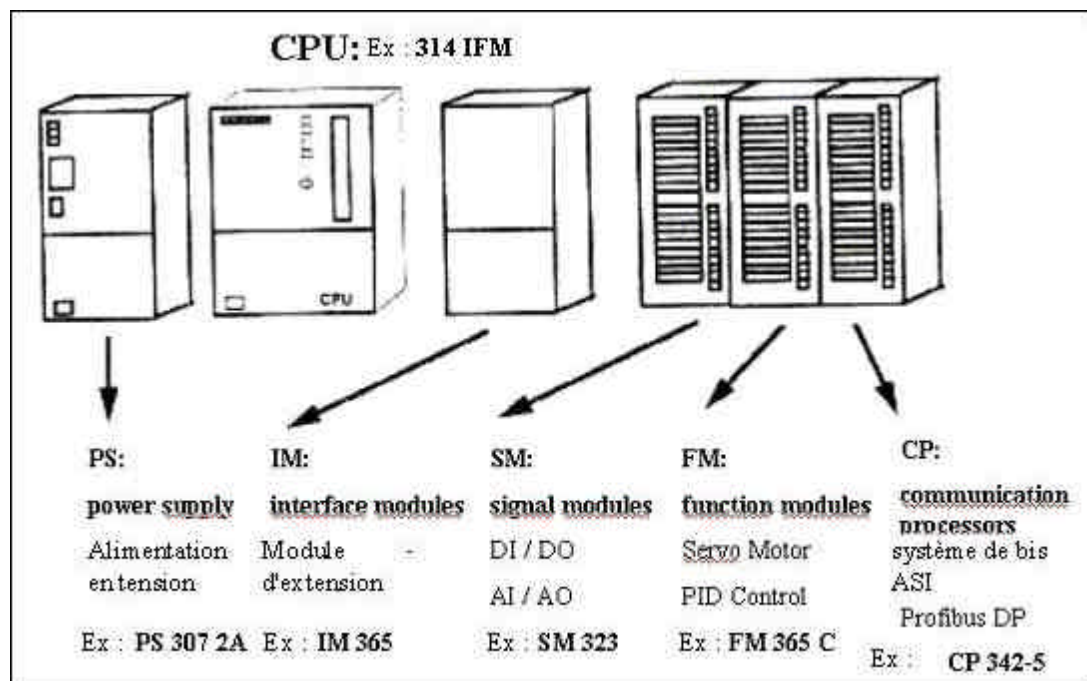
6. INSTALLATION ET MISE EN ŒUVRE DU SIMATIC S7-300



Gamme de modules :

Le SIMATIC S7-300 est un système d'automatisation modulaire offrant la gamme de modules suivants :

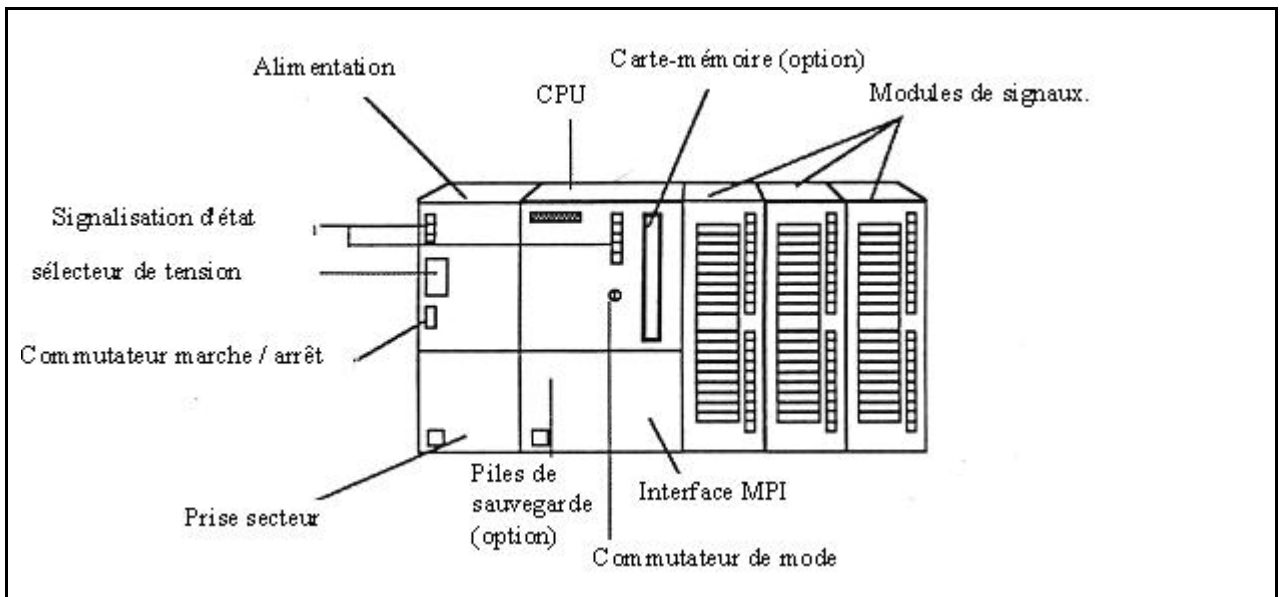
- Unités centrales (CPU) de capacités différentes, certaines avec d'entrées/sorties intégrées (ex : CPU314C) ou avec interface PROFIBUS intégrée (ex : CPU315-2DP)
- Modules d'alimentation PS avec 2A, 5A ou 10A
- Modules d'extension IM pour configuration de plusieurs lignes du SIMATIC S7-300
- Modules de signaux SM pour entrées et sorties numériques et analogiques
- Modules de fonction FM pour fonctions spéciales (ex : pilotage d'un moteur pas à pas)
- Processeurs de communication CP pour la connexion au réseau



Indications : Pour ce module, vous aurez seulement besoin d'un module d'alimentation, d'une CPU quelconque et d'un module d'entrées et de sorties numériques.



Éléments principaux de la tension d'alimentation et de la CPU :



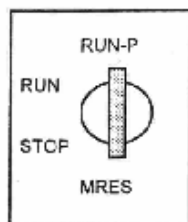
Interface MPI :

Chaque CPU est équipée d'une interface MPI pour la connexion de la ligne de programmation (ex : PC Adapter).

Celle-ci se trouve sous le volet de protection à l'avant de la CPU.

Commutateur de mode de fonctionnement :

Chaque CPU est équipée d'un commutateur de mode permettant de changer de mode de fonctionnement. Celui-ci est principalement un commutateur à clé amovible qui permet de basculer dans les modes de fonctionnement MARCHE (RUN) et ARRÊT (STOP). Les modes de fonctionnement suivants sont possibles :



RUN-P :	Exécution du programme ; toutes les fonctions PG sont autorisées
RUN :	Exécution du programme ; seules les fonctions PG lecture sont autorisées
STOP :	Le programme n'est pas exécuté ; toutes les fonctions PG sont autorisées
MRES :	Position dans laquelle un effacement général peut être effectué



Indications : Le commutateur pour les CPU de la série CPU31xC est un interrupteur à bascule. Il manque alors ici le mode de fonctionnement Marche avec protection en écriture. Les fonctions PG écrites sont permises en mode Marche.



Effacement général :

Cette fonction efface toutes les données utilisateur sur la CPU mais cependant pas la Memory Card ou la Micro Memory Card. L'effacement général doit être exécuté une fois avant le début de la programmation.

Ceci a lieu en trois étapes :

Etape	Action	Résultat
1	Mettez le commutateur de mode de fonctionnement en position STOP .	Le témoin STOP s'allume
2	Mettez le commutateur en position MRES et maintenez-le dans cette position (environ 3 secondes) jusqu'à ce que le témoin STOP se rallume	Le témoin STOP s'éteint et après environ 3 secondes, s'allume à nouveau. Pour les nouvelles CPU, attendez que le témoin STOP s'allume une seconde fois. Important : Entre les étapes 2 et 3, il doit s'écouler au maximum 3 secondes.
3	Remettez le commutateur en position STOP et ensuite remettez en position MRES après un délai de 2 secondes	Le témoin STOP clignote pendant environ 3 secondes et se rallume ensuite : Tout est o.k.; l'effacement général de la CPU a été exécuté

7. EXEMPLE D'APPLICATION



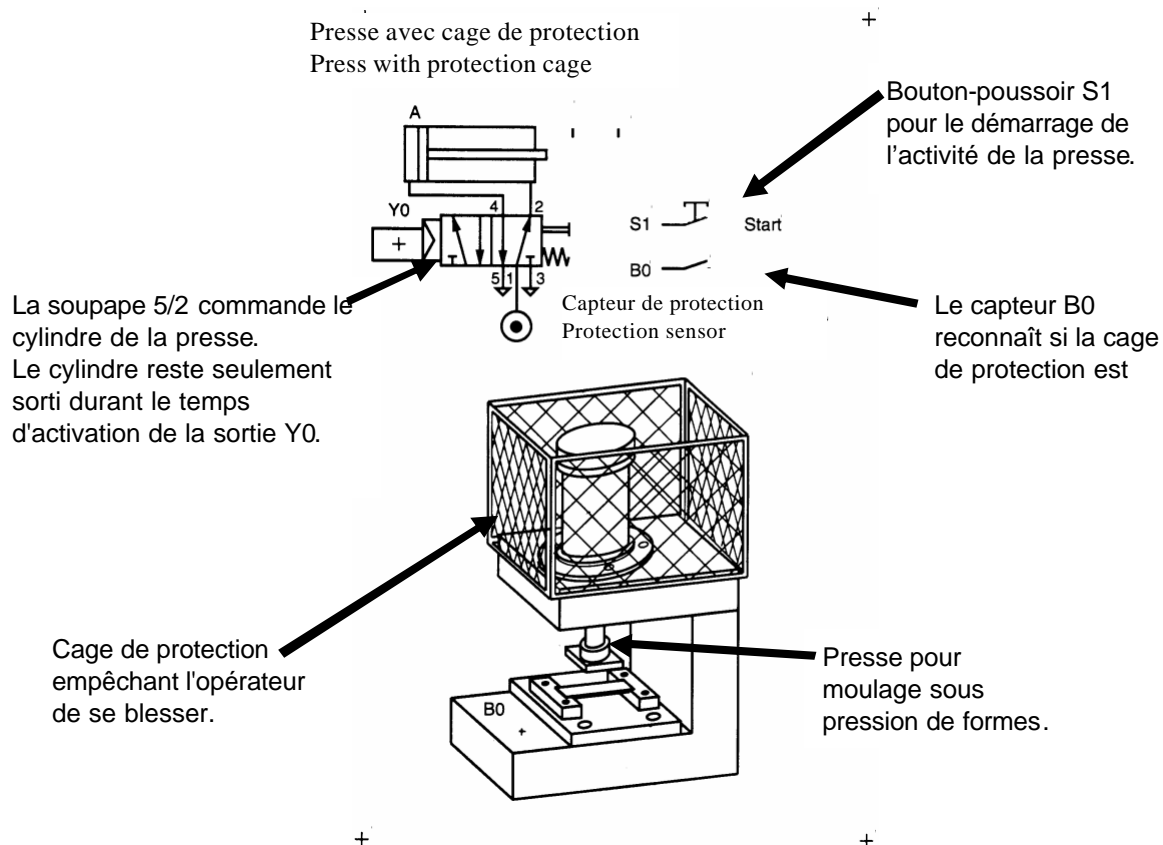
Nous allons dans notre premier programme STEP 7 résoudre une tâche simple.

Une presse avec cage de protection ne doit être déclenchée avec le bouton-poussoir de démarrage S1 que si la cage de protection est fermée. Cet état est surveillé par un capteur de cage de protection B0. Si c'est le cas, une soupape 5/2 Y0 pour le cylindre de presse est activée pendant exactement 10 secondes pour le moulage sous pression de la forme.

Pour des raisons de sécurité, la presse doit remonter lorsque le bouton-poussoir S1 est relâché ou que le capteur cage de protection B0 ne réagit pas.

Liste d'attribution :

Adresse	Mnémonique	commentaire
E 0.0	B0	Capteur cage de protection
E 0.1	S1	Bouton poussoir de démarrage
A 4.0	Y0	Soupape 5/2 pour le cylindre de presse



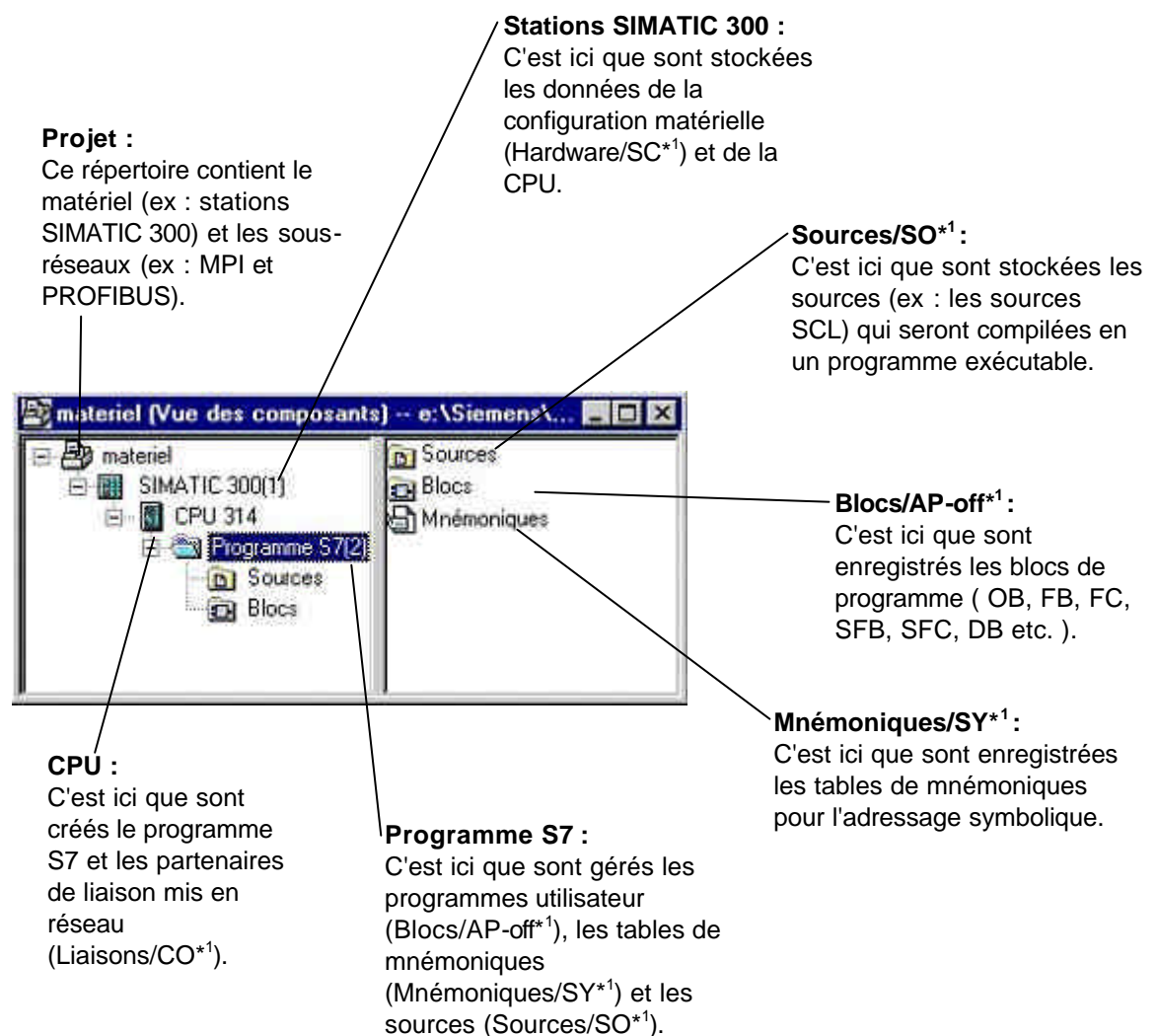
8. CREATION DU PROJET STEP 7



La gestion de fichiers a lieu dans STEP 7 avec '**SIMATIC Manager**'. Là, vous pouvez d'un clic de la souris par exemple copier des blocs de programme ou appeler d'autres outils pour en poursuivre le traitement. L'interface utilisateur est conforme au standard WINDOWS 95/98/2000/ME/NT4.0/XP habituel (un clic sur le bouton droit de la souris ouvre par exemple un menu pour le composant sélectionné).

La structure matérielle de l'automate programmable est reproduite dans les répertoires '**Station SIMATIC 300**' et '**CPU**'. Un tel projet est par conséquent toujours spécifique au matériel.

Chaque projet est créé dans STEP 7 avec une structure prédéfinie. Les programmes sont enregistrés dans les répertoires suivants :



*¹ Désignations de la version 2.x de STEP 7

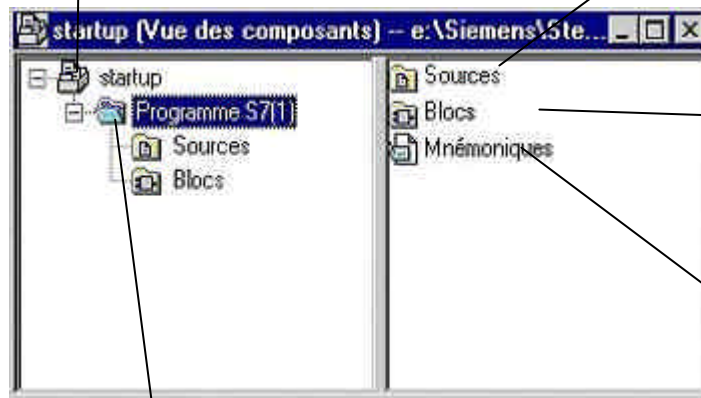


Il est toutefois possible de créer un projet sans ces objets indépendamment d'une configuration matérielle donnée.

Il s'ensuit la structure suivante :

Projet :

Ce répertoire contient le matériel (ex : stations SIMATIC 300) et les sous-réseaux (ex : MPI et PROFIBUS).



Sources/SO*¹ :

C'est ici que sont stockées les sources (par ex. les sources SCL) qui seront compilées en un programme exécutable.

Blocs/AP-off*¹ :

C'est ici que sont stockés les blocs de programme (OB, FB, FC, SFB, SFC, DB etc.).

Mnémoniques/SY*¹ :

C'est ici que sont stockées les tables de mnémoniques pour l'adressage symbolique.

Programme S7 :

C'est ici que sont gérés les programmes utilisateur (Blocs/AP-off*¹), les tables de mnémoniques (Mnémoniques/SY*¹) et les sources (Sources/SO*¹).

*¹ Désignations de la version 2.x de STEP 7



Indications : Cet exemple est créé sans configuration du matériel. Les programmes peuvent donc être chargés dans n'importe quelle configuration SIMATIC S7-300, S7-400 ou WinAC. Seules les adresses des entrées et sorties devront le cas échéant être modifiées.



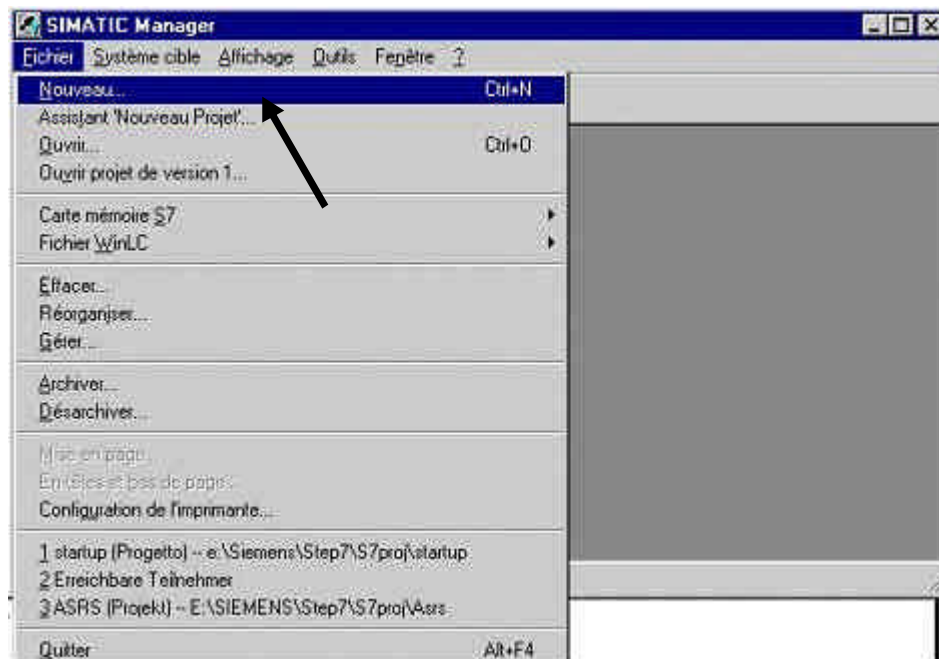
Les étapes suivantes sont à exécuter par l'utilisateur pour créer un projet dans lequel le programme permettant de résoudre la tâche sera écrit.

1. L'outil central dans STEP 7 est '**SIMATIC Manager**', qui est appelé par un double-clic. (→ SIMATIC Manager)



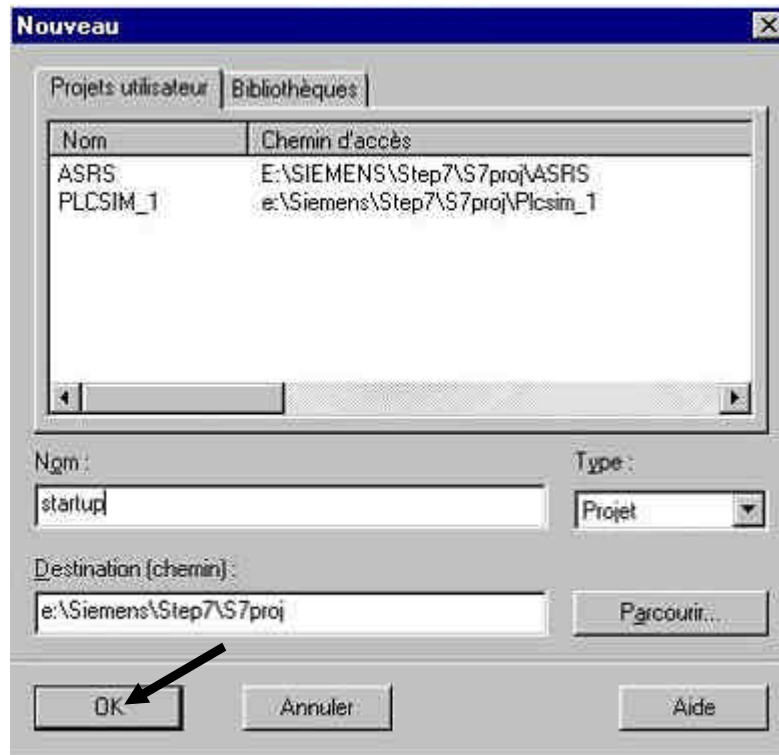
SIMATIC Manager

2. Les programmes de STEP 7 sont gérés dans des projets. Nous allons donc commencer à créer un projet (→ Fichier → Nouveau)

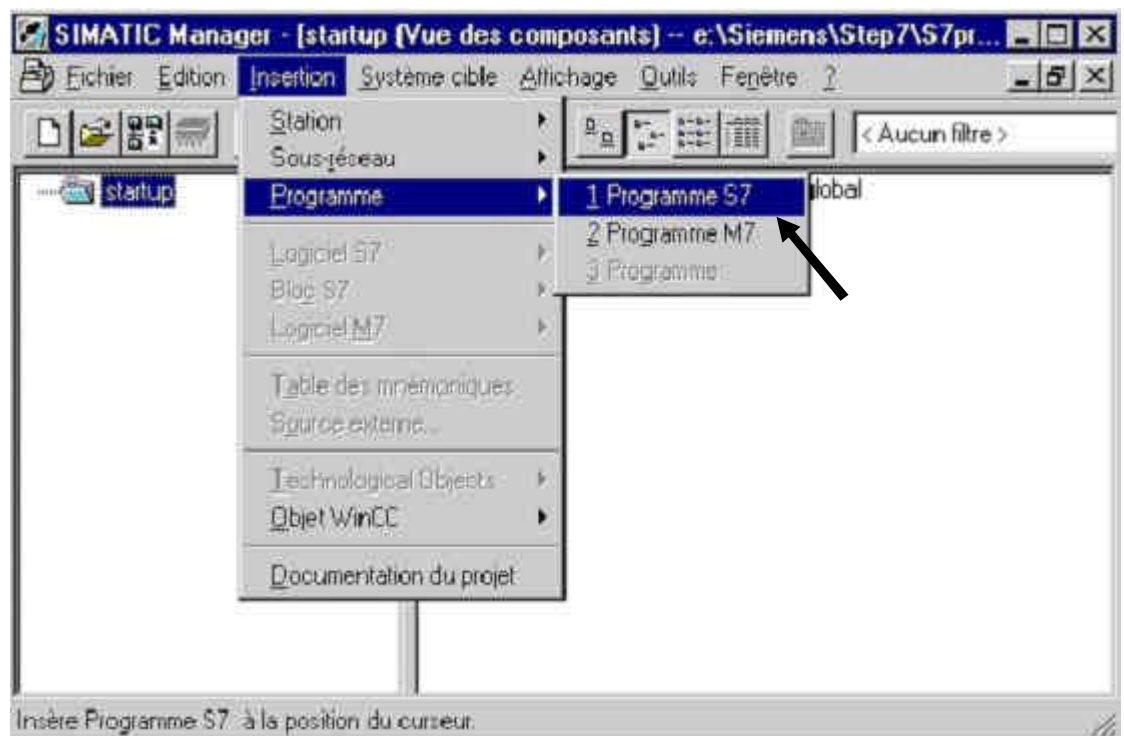




3. Donnez au projet le **,Nom' de ,Startup'.** (→ Startup → OK)



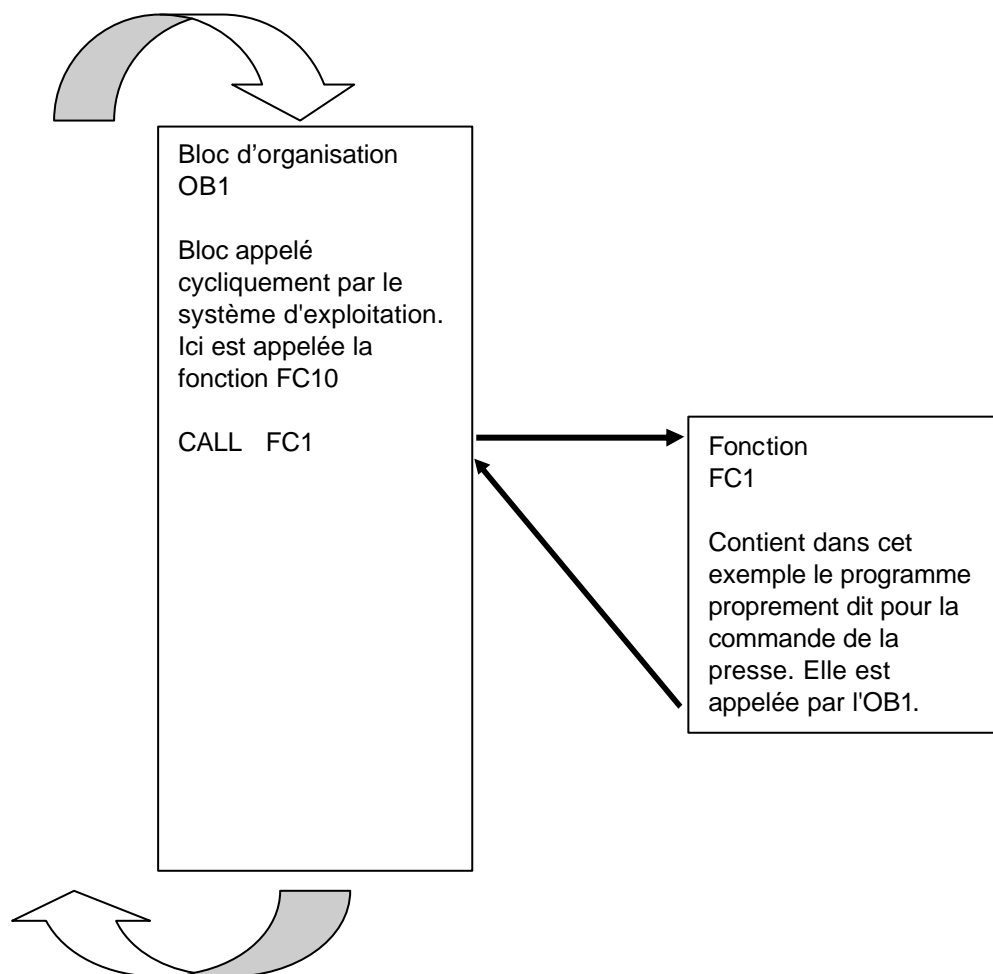
4. Ensuite, insérez dans le projet **,Startup'** un nouveau **,Programme S7'.** (→ Startup → Insertion → Programme → Programme S7)





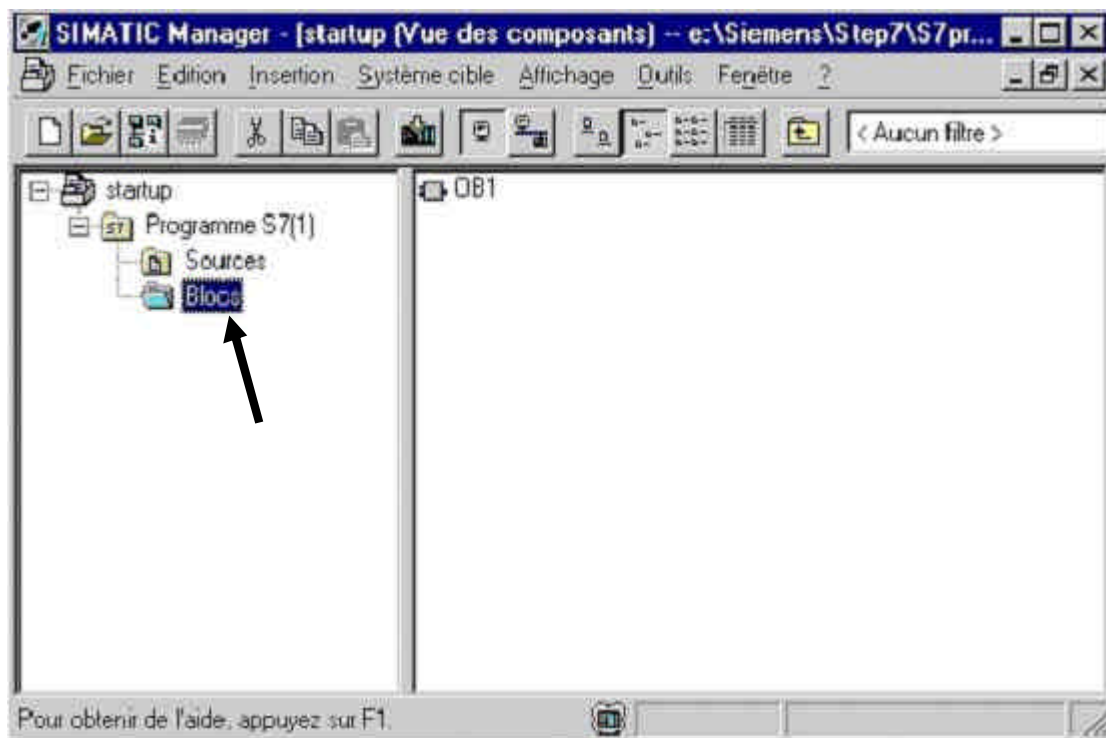
5. La séquence du programme est écrite dans STEP 7 au moyen de blocs. Le bloc d'organisation OB1 y figure par défaut. Celui-ci constitue l'interface avec le système d'exploitation de la CPU et est automatiquement appelé par ce dernier et traité cycliquement. Dans ce bloc d'organisation sont ensuite appelés d'autres blocs comme par exemple la fonction FC1. C'est ce qu'on appelle la programmation structurée. Elle permet de décomposer une tâche en plusieurs tâches partielles qui sont ainsi plus faciles à résoudre et à tester dans leur fonctionnalité.

Structure du programme de l'exemple :



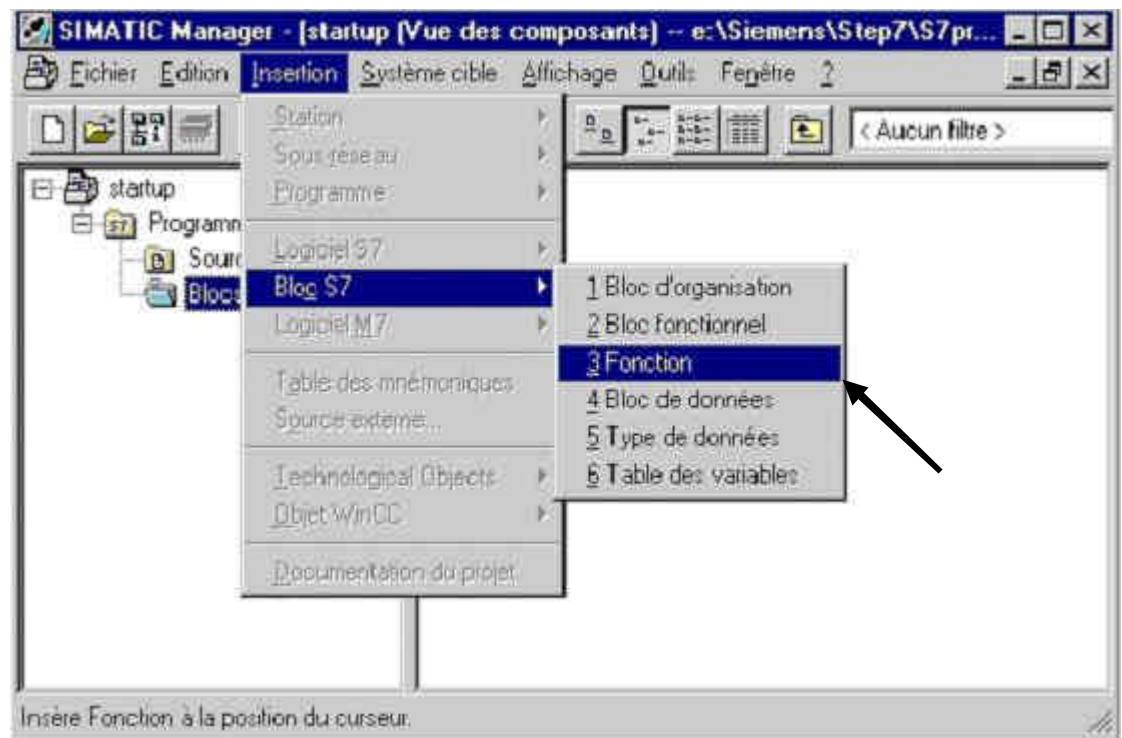


6. Sélectionnez le dossier 'Blocs' pour insérer un nouveau bloc FC1 dans le projet. (→ Blocs)

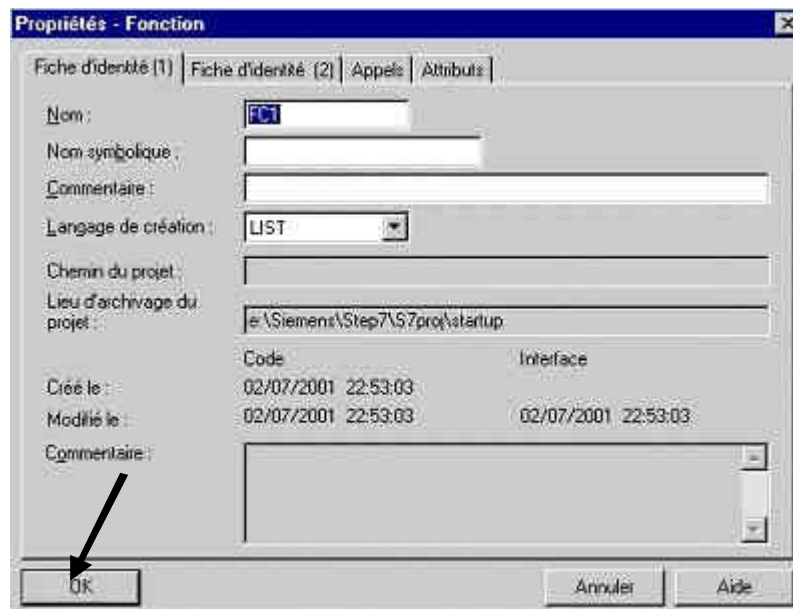




7. Le '**bloc S7**' : '**Fonction**' est alors inséré dans le dossier Blocs. (→ Insertion → Bloc S7 → Fonction)

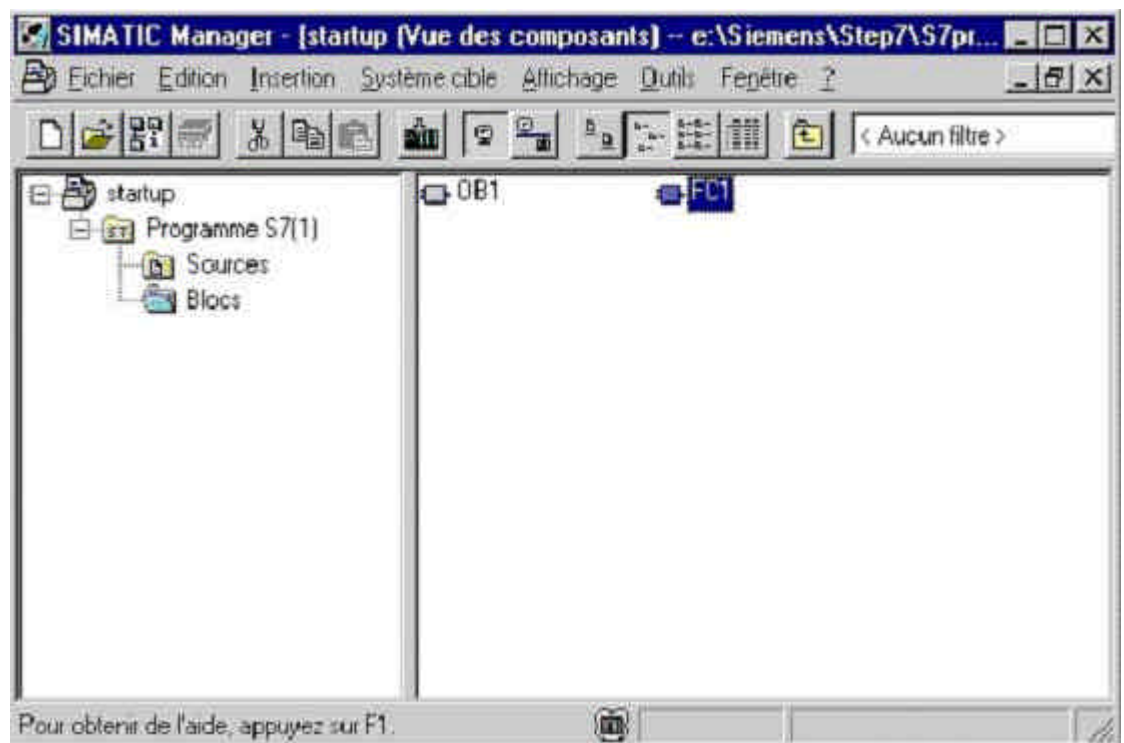


8. Il est maintenant possible d'entrer un nom pour la fonction et de sélectionner diverses options pour la documentation des blocs. (→ FC1 → OK)





9. Les deux blocs OB1 et FC1 sont à présent insérés dans SIMATIC Manager pour servir ultérieurement à la programmation.

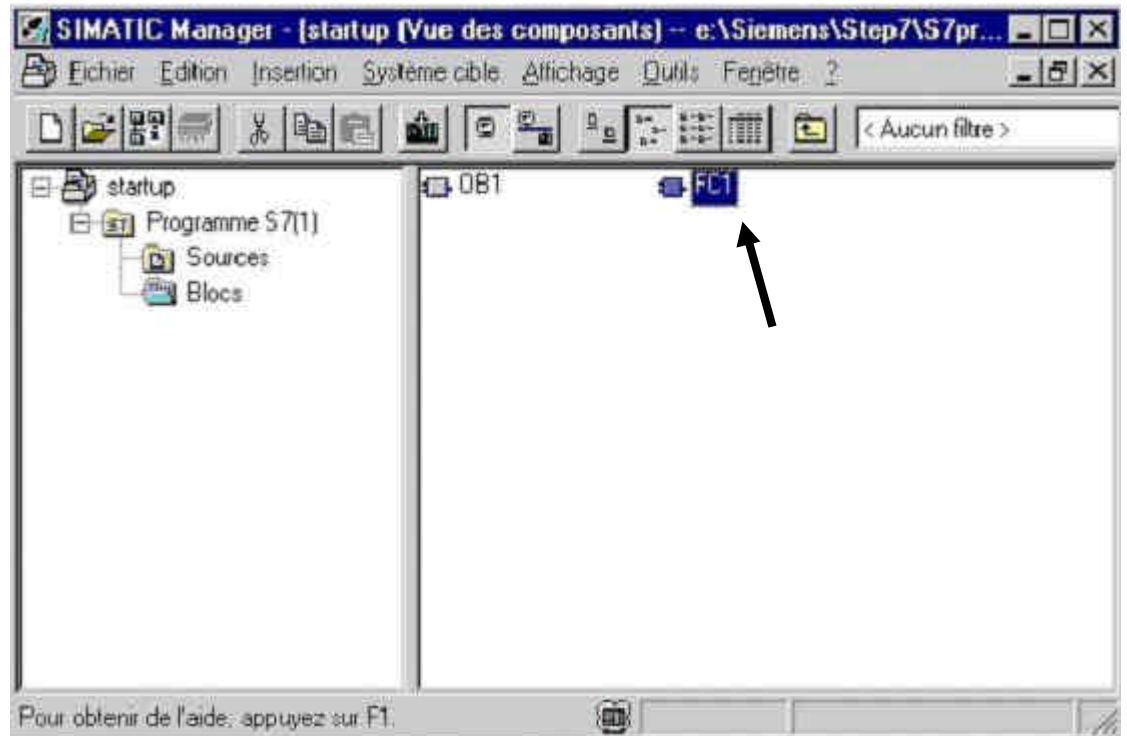


9. ECRITURE DU PROGRAMME STEP 7 DANS LE LOGIGRAMME LOG

LOG (Logigramme) est l'un des éditeurs graphiques de STEP 7 dans lequel il est possible de créer le programme S7. Dans ce langage de programmation, des boîtes symbolisent les différentes fonctions de la tâche d'automatisation. A gauche de chaque symbole se trouvent les entrées, à droite les sorties.

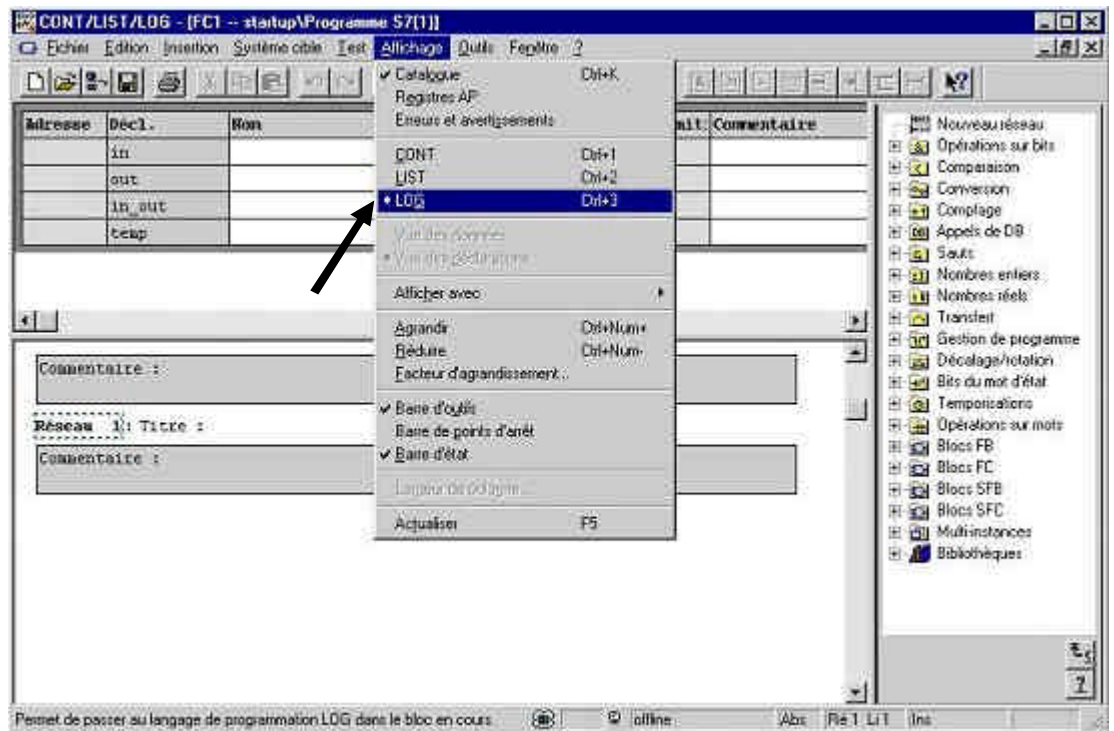


1. Le premier bloc que nous éditerons est la fonction '**FC1**'. Pour l'ouvrir, double-cliquez sur son icône dans '**SIMATIC Manager**' (→ FC1)





- Sélectionnez dans l'éditeur de programmes '**CONT/LIST/LOG**' , le langage de programmation Logigramme '**LOG**' dans le menu '**Affichage**'.(→ Affichage → LOG)





3. L'interface de programmation pour une programmation en logigramme (LOG) est de la manière suivante :

Commandes fréquemment utilisées telles que boîte ET, boîte OU, affectation, entrée binaire, négation de l'entrée binaire, branchement T, connexion

Sauvegarde du bloc

Chargement du bloc dans la CPU

Insertion d'un nouveau réseau

Catalogue complet des éléments de programme

Table de déclaration des variables (n'est pas utilisée dans cet exemple)

Champs de commentaires et titres de réseaux et de blocs

C'est ici que vous programmez la tâche à l'aide des éléments graphiques LOG

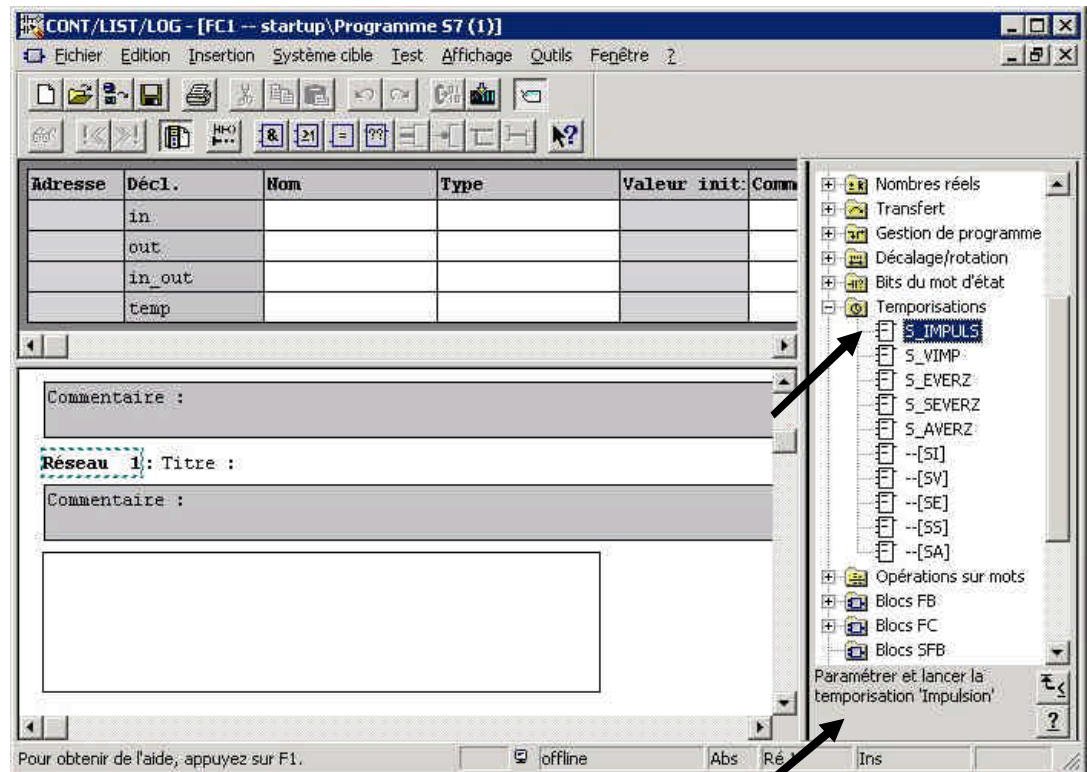
Vous faites glisser les éléments de programme avec la souris dans le réseau. Il ne vous reste plus qu'à entrer vos opérandes.



Indications : Les programmes dans les blocs de STEP 7 sont développés dans des réseaux séparés. Ceci permet de créer de larges structures et permet une meilleure documentation des titres de réseaux.



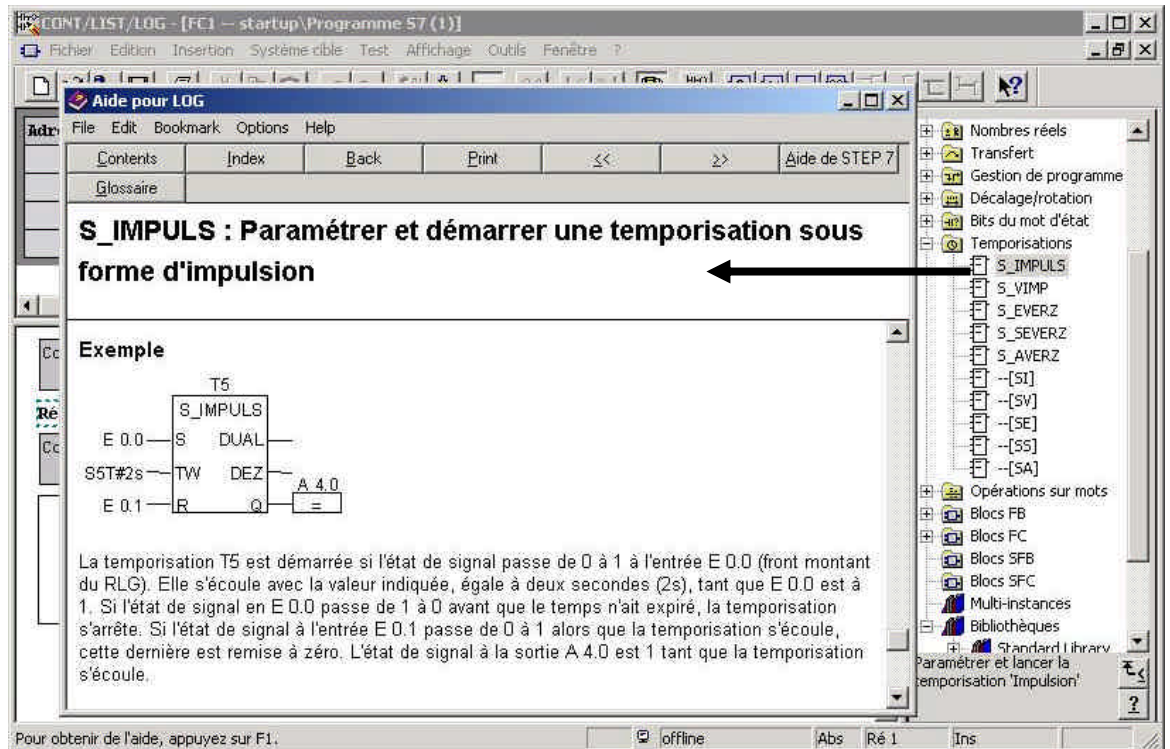
4. Nous avons besoin pour notre exemple d'une temporisation sous forme d'impulsion. Elle s'appelle dans STEP 7 **'S_IMPULS'** et figure dans le catalogue sous le répertoire **'Temporisations'**. (→ Temporisations → S_IMPULS)



Indications : Lorsqu'une opération est sélectionnée, une brève information à propos de cette opération s'affiche au bas du catalogue.



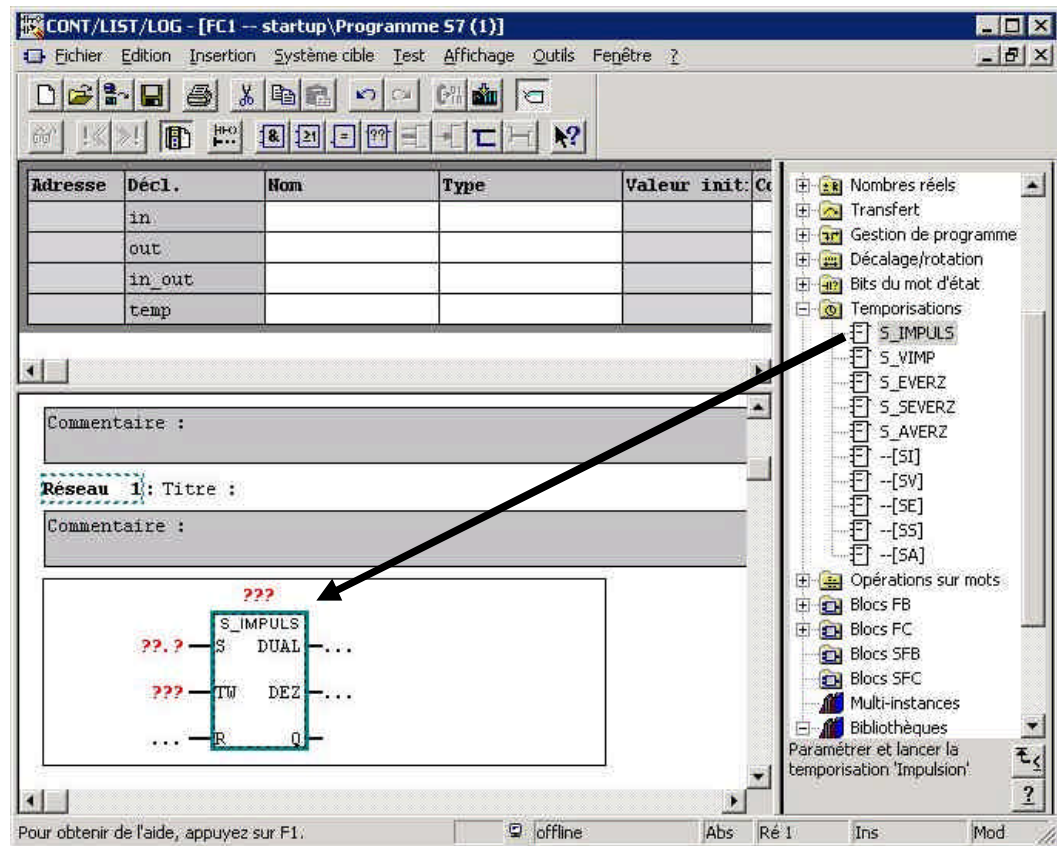
5. Pour obtenir une description plus détaillée sur l'opération, cliquez sur le bouton **F1** pour afficher l'aide en ligne correspondante dans laquelle chaque commande est expliquée à l'appui d'un exemple. (→ **F1**)





Indications : La temporisation sous forme d'impulsion **'S_IMPULS'**, utilisée ici, délivre '1' à la sortie **'Q'** pour la durée définie si l'entrée SET **'S'** est égale à '1'. La sortie **'Q'** est remise à '0' si le temps prédéfini sous **'TW'** est écoulé ou si le niveau du signal de l'entrée SET **'S'** est à nouveau '0'.

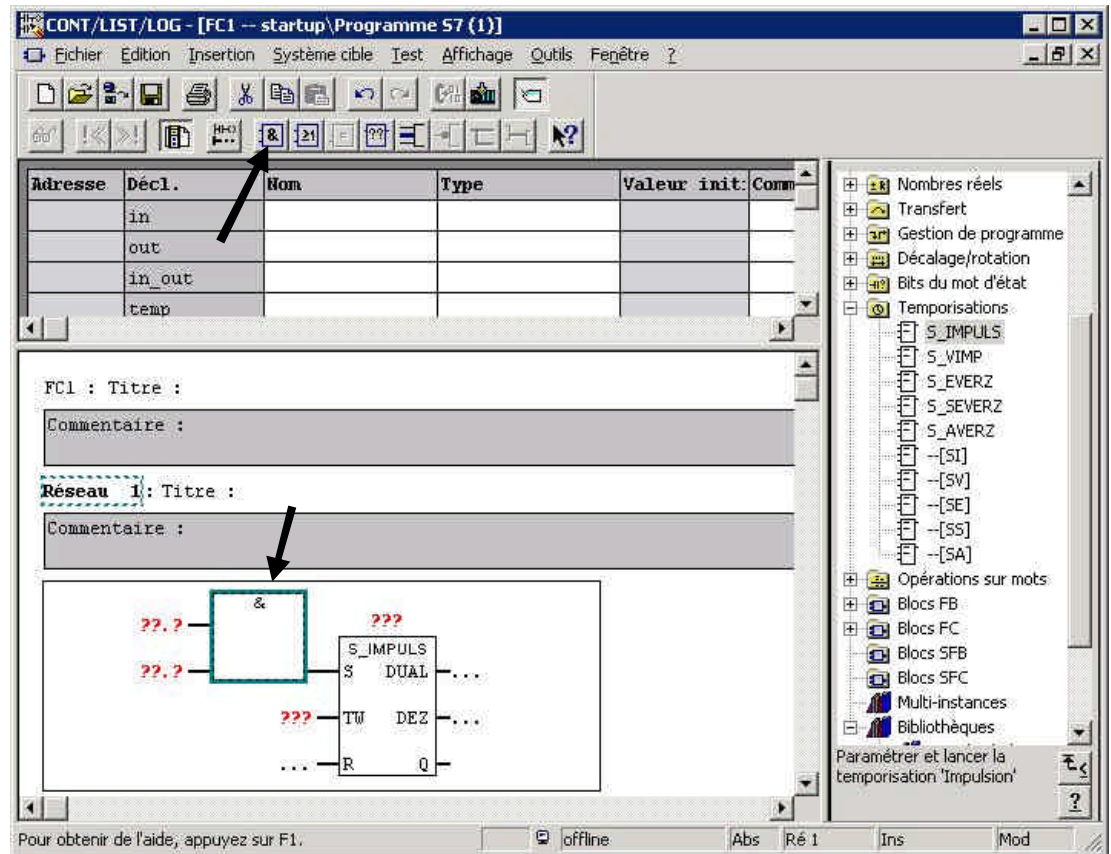


6. Insérez dans le premier réseau l'opération 'S_IMPULS' après l'avoir sélectionnée dans le catalogue en maintenant le bouton de la souris enfoncé et en le relâchant une fois dans le réseau.(→ S_IMPULS)



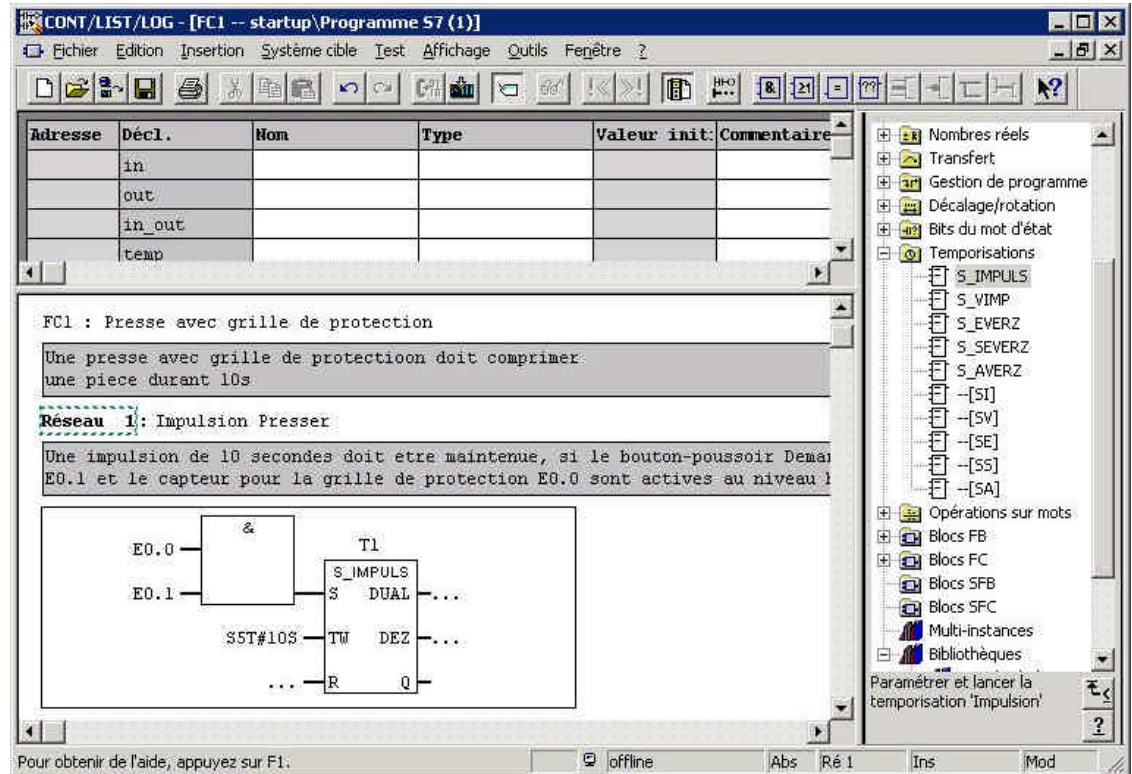


7. Les opérations fréquemment utilisées telles que la fonction ET se trouvent également dans la barre d'outils. Pour insérer celle-ci, cliquez d'abord sur l'entrée '**S**' de la temporisation, puis sur l'icône . (→ S → )





8. Nommez maintenant la temporisation, **T1** et entrez une valeur temporelle de 10 secondes en format S5Time : **'S5T#10s'**. De plus, complétez les entrées '**E 0.0**' et '**E 0.1**' de la fonction ET et entrez un commentaire pour le réseau et le bloc. (→ T1 → S5T#10s → E0.0 → E0.1 → commentaires)



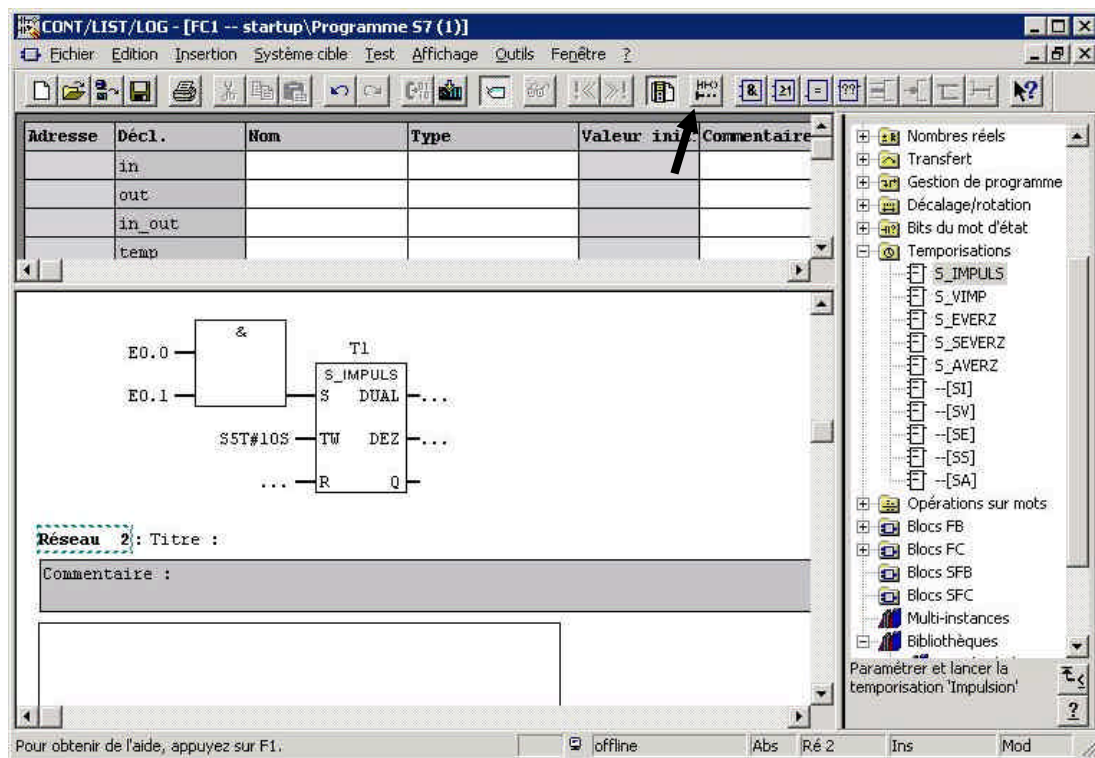
Indications : La saisie du temps pour une temporisation doit respecter la syntaxe suivante :

S5T# 10s

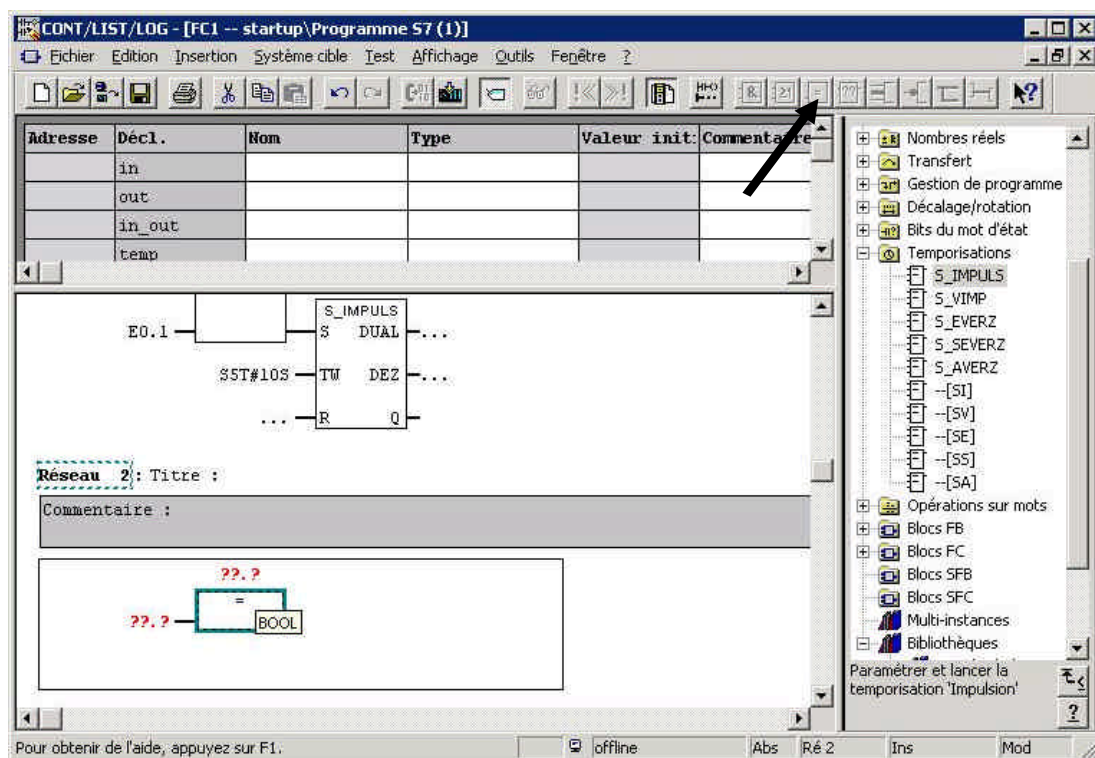
Saisissez d'abord **S5T#** pour le format puis directement le temps voulu en secondes (ici **10 Secondes**). Il est également possible d'entrer des millisecondes (MS), des minutes (M) et des heures (H). Il est également possible de combiner ces unités (ex : S5T#3M_3S).







9. Insérez ensuite un nouveau réseau en cliquant sur l'icône ,  , dans la barre d'outils. (→ )

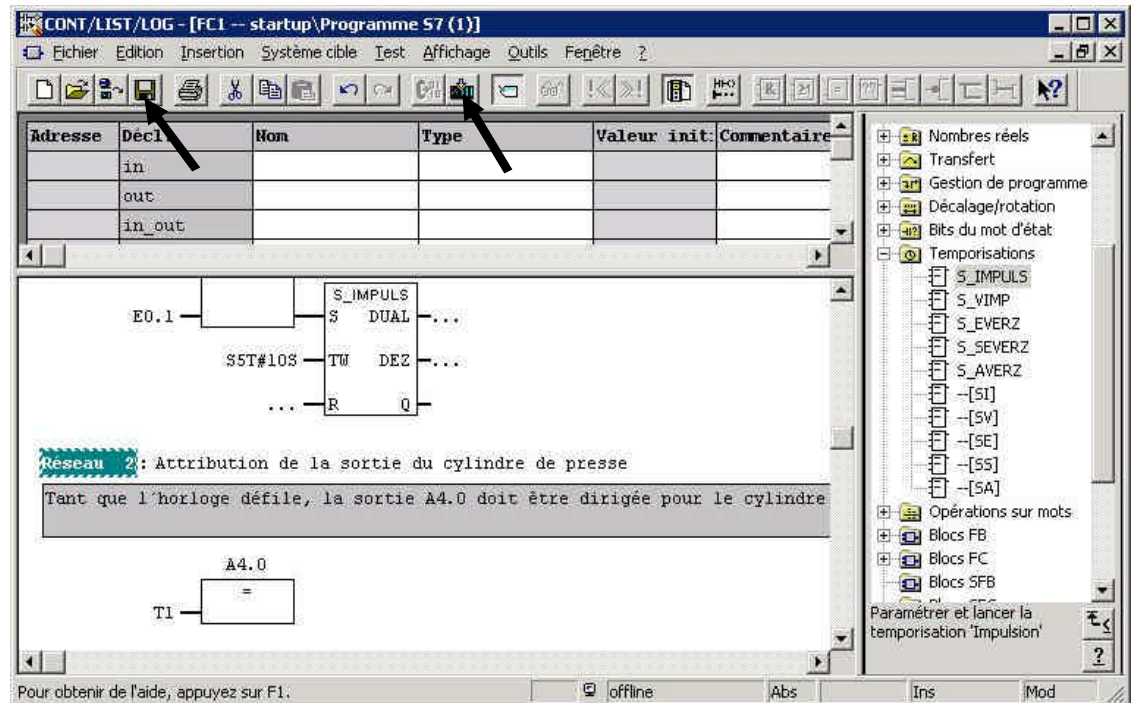


10. Cliquez ensuite sur l'icône ,  , pour insérer une affectation (→ )



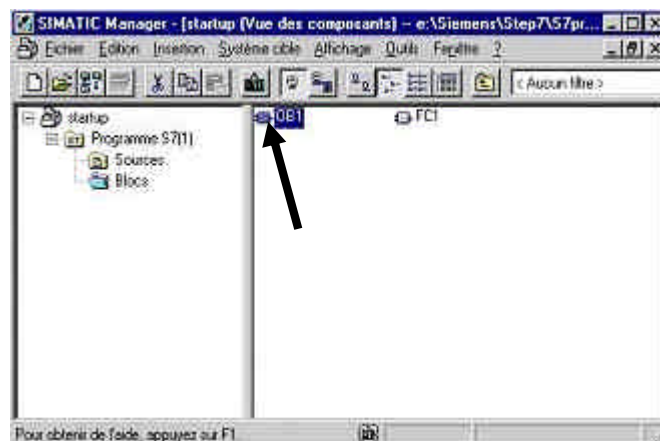


11. L'affectation s'applique à la sortie '**A4.0**' et vaut pour la durée où le signal '**T1**' est égal à "High".
Insérez encore ces deux opérands avant d'enregistrer,  FC1 et de le charger dans l'automate programmable, . (→ A4.0 → T1 →  → )



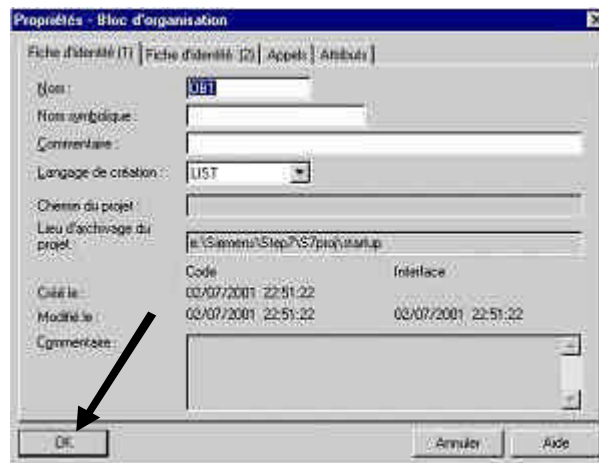
- Indications :**
- Ne fermez pas l'éditeur de programmes „CONT/LIST/LOG“.
 - Ou bien, vous commutez en bas de page sur SIMATIC Manager (Point 12)
 - Ou bien, vous appelez OB1 avec la fonction „OUVRIR“.

12. Pour programmer l'appel de la fonction FC, ouvrez le bloc '**OB1**' dans '**SIMATIC Manager**' par double clic (→ SIMATIC Manager → OB1)

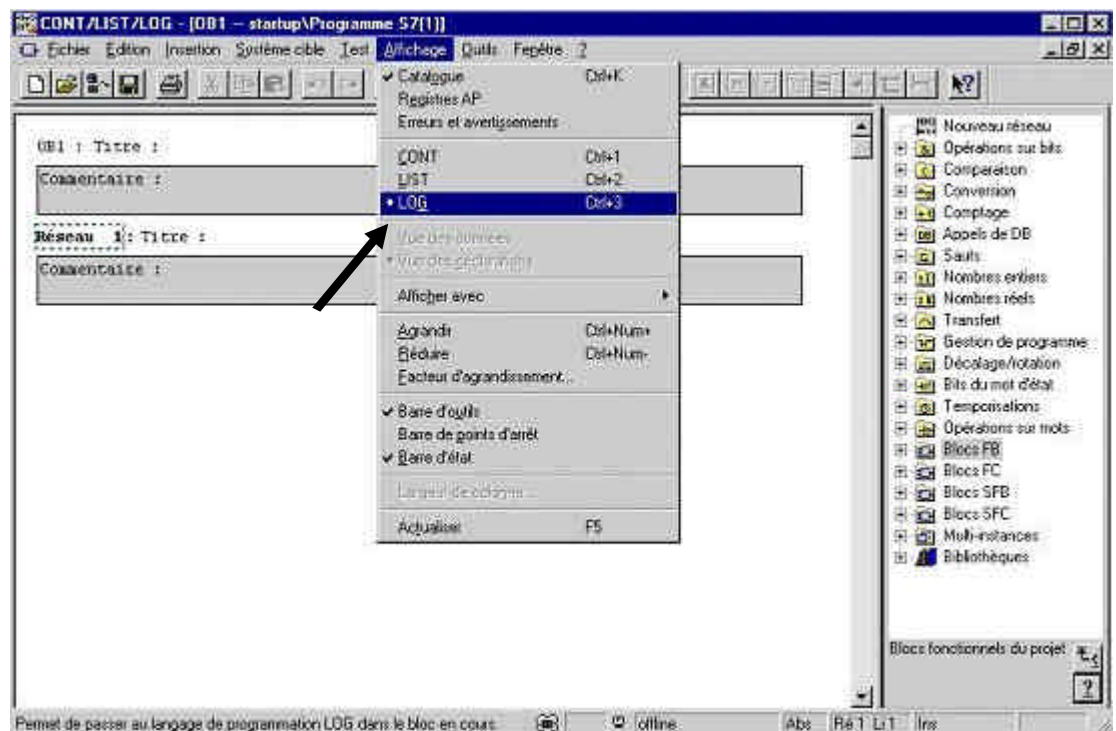








13. Conservez les propriétés de OB1 et validez avec **,OK'** (→ OK)

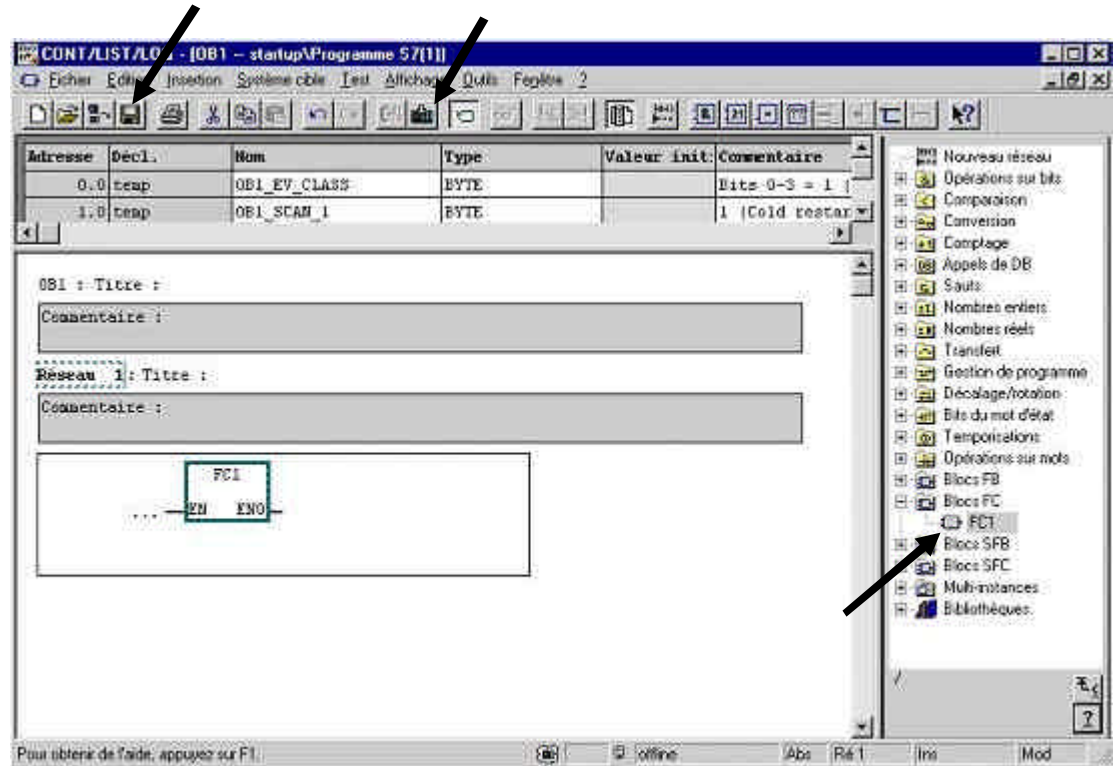


14. Sélectionnez dans l'éditeur de programmes **'CONT/LIST/LOG'** le langage de programmation **'LOG'** dans le menu **'Affichage'**. (→ Affichage → LOG)





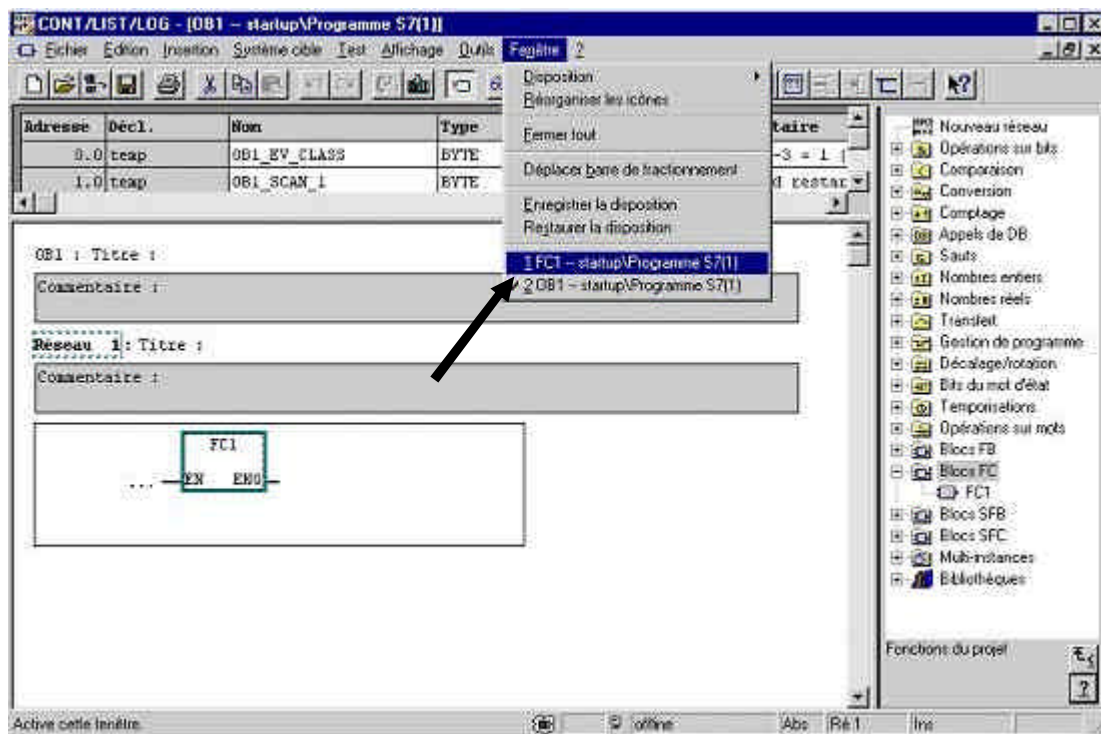
15. Insérez la '**FC1**' qui se trouve dans le dossier '**Blocs FC**' du catalogue en double-cliquant avec la souris dans le réseau 1 du bloc OB1. Enregistrez ensuite OB1,  et chargez ce bloc dans l'automate programmable, . (→ Blocs FC → FC1 →  → )





10. TEST DU PROGRAMME STEP 7 DANS LA CPU



1. Pour visualiser le programme du FC1, sélectionnez dans l'éditeur de programmes 'CONT/LIST/LOG' le menu **'Fenêtre'** puis **'FC1'**. (→ Fenêtre → FC1)





2. Un clic de la souris sur l'icône avec des lunettes, , lance la visualisation du programme de FC1. Vous pouvez alors voir le temps de la temporisation s'écouler et l'état logique des entrées et sorties changer. (→ )

